

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах**

«На правах рукопису»
УДК 004.031.6

До захисту допущено:
Завідувач кафедри
_____ Олександр РОЛІК
«__» _____ 20__ р.

**Магістерська дисертація
на здобуття ступеня магістра
за освітньо-професійною програмою «Програмне забезпечення інформаційно-
комунікаційних систем»
зі спеціальності 121 «Інженерія програмного забезпечення»
на тему: «Система підтримки життєдіяльності рослин у теплиці»**

Виконав:
студент VI курсу, групи ІТ-93мп
Рещенко Євген Юрійович _____

Керівник:
Доцент каф. АУТС, к.т.н., с.н.с.
Кравець Петро Іванович _____

Рецензент:
Доцент каф. ТК, к.т.н., с.н.с.
Тимошин Юрій Афанасійович _____

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.
Студент _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра автоматики та управління в технічних системах

Рівень вищої освіти – другий (магістерський)

Спеціальність – 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма «Програмне забезпечення інформаційно-комунікаційних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

_____ Олександр РОЛІК

«___» _____ 20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Рещенко Євген Юрійович

1. Тема дисертації «Система підтримки життєдіяльності рослин у теплиці», науковий керівник дисертації доцент каф. АУТС, к.т.н., с.н.с. Кравець Петро Іванович, затверджені наказом по університету від «26» 10 2020 р. №3132-с

2. Термін подання студентом дисертації 17. 12. 2020

3. Об'єкт дослідження: автоматизація процесу зрошення рослин

4. Предмет дослідження: система автоматизації процесу зрошення рослин

5. Перелік завдань, які потрібно розробити: розробка моделі системи, що задовольнить потреби підтримки параметрів важливих для оптимального росту рослини, розробка структурної та функціональної реалізації системи, створення програмного коду, технічна реалізація та запуск для тестування, тестування програмного забезпечення для апаратної частини

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: Діаграма класів, Алгоритм, Діаграма екранних форм, Структура взаємодії елементів системи, Діаграма прецедентів, Схема камери для замішування, Структурна схема, функціональна схема

7. Дата видачі завдання: 01.09.2020

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
	Отримання завдання	01.09.2020	
	Опрацювання літературних джерел	05.09.2020 – 20.09.2020	
	Аналіз предметної області	21.09.2020 – 01.09.2020	
	Проектування системи	02.10.2020 – 21.10.2020	
	Розробка системи	22.10.2020 – 22.11.2020	
	Розробка стартап-проекту	23.11.2020 – 30.11.2020	
	Оформлення дисертації	01.12.2020 – 15.12.2020	
	Захист дисертації	23.12.2020	

Студент

Євген РЕЩЕНКО

Науковий керівник

Петро КРАВЕЦЬ

РЕФЕРАТ

Магістерська дисертація на здобуття ступеня магістру на тему “ Система підтримки життєдіяльності рослин у теплиці ”: 101с., 23 рис., 26 табл., 8 додатки, 21 джерел.

Об'єкт дослідження – автоматизація процесу зрошення рослин.

Мета роботи – розробка системи, що забезпечить можливість покращеного зрошення рослин шляхом підтримки ідеальних параметрів середовища з мінімальним втручанням людини.

У магістерській дисертації розглянуто проблему автоматизації зрошення рослин. Отримано перелік параметрів, зміну та підтримку яких система має автоматизувати. Отримано уяву про взаємодію елементів системи з можливими елементами, зображеними у додатку Д. Також спроектовано діаграму прецедентів, зображену у додатку Ж.

Визначено основні елементи органів системи. Побудована структура на основі якої побудовано функціональну схему. Описано взаємодію кожного елемента для кожного органу системи, їх призначення, методи взаємодії.

Створено алгоритм, за яким система має змінювати параметри суміші. На основі структурної, функціональної схеми, алгоритму та технічного завдання спроектовано програмне забезпечення для мікроконтролера. Розроблено механізм зберігання сценаріїв. Створено «захоплюючий портал» для первинного підключення пристрою до мережі Інтернет. Отримано програмне забезпечення готове до завантаження у МК.

Отримано робочу систему. Отримано результати роботи системи, що задовольняють потреби. Система підтримує параметри кислотності у границях «задана – 0.5 од» < задана < «задана + 0.5 од», що є допустимим для проходження росту рослини, подає потрібні кількості добрив та кількість чистої води.

Проведено аналіз ринку та розробка стратегії запуску стартап-проєкту. враховано всі ризики, маркетингові стратегії та ключові переваги концепції потенційного продукту, побудовано бізнес-план подальшого розвитку системи.

Ключові слова: ПРОГРАМУВАННЯ МІКРОКОНТРОЛЕРА, ІНТЕРНЕТ РЕЧЕЙ, АВТОМАТИЗАЦІЯ, ЗРОЩУВАННЯ РОСЛИН.

ABSTRACT

Master's thesis for master's degree on the topic " Plant's life support system in greenhouse": 101p, 23 figures, 26 tables, 8 annexes, 21 sources.

Object of study – plant's grow process automation.

The purpose of the work – develop plant automation system to obtain better growing experience by maintaining ideal plant's environment parameters. Also the goal is to free user from maintaining this parameters manually.

In the master's thesis the problem of automation of plant growing is considered. The list of changing parameters which the system should automate is determined. An idea of the interaction of the elements of the system with the possible elements determined. The diagram of precedents is also designed.

Identified the main elements of the system. The interaction of each element for each unit of the system, their purpose, methods of interaction are described.

An algorithm has been created according to which the system must change the parameters of the mixture. Based on the structural, functional scheme, algorithm and technical task, the software for the microcontroller is designed. A mechanism for storing scripts has been developed. A "captive portal" has been created for the initial connection of the device to the Internet.

Got working system able to be tested. The results of the system operation that meet the needs are obtained. The system maintains the acidity parameters within the limits: "set – 0.5 units" < set < "set + 0.5 units", which is acceptable for the growth of the plant, provides the required amount of fertilizers and the amount of clean water.

The market analysis and development of the strategy of launching a startup project are carried out. All risks, marketing strategies and key advantages of the potential product concept are taken into account, a business plan for further development of the system is built.

Keywords: MICROCONTROLLERS PROGRAMMING, INTERNET OF THINGS, AUTOMATION, PLANTS GROWING.

ЗМІСТ

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ.....	9
ВСТУП.....	11
1 ОГЛЯД ТЕХНОЛОГІЙ І СИСТЕМ АВТОМАТИЗАЦІЇ.....	15
1.1 Змістовний опис і аналіз предметної області	15
1.2 Аналіз існуючих рішень	20
1.3 Можливість використання у аграрній промисловості	24
1.4 Висновки до розділу	28
2 РЕАЛІЗАЦІЯ НА СТРУКТУРНОМУ ТА ФУНКЦІОНАЛЬНОМУ РІВНІ.....	29
2.1 Структура системи.....	29
2.1.1 Керуючий орган	29
2.1.2 Регулюючий орган	30
2.2 Функції елементів системи.....	31
2.2.1 Функції мікроконтролера	32
2.2.2 Функції МЗ	32
2.2.3 Функції МФК та ГРЧ	32
2.2.4 Функції МКСК	32
2.3 Взаємодія елементів електронної схеми.....	33
2.4 Режим роботи ESP8266 та Mega2560	35
2.5 Висновки до розділу	35
3 РЕАЛІЗАЦІЯ АЛГОРИТМІЧНОГО І ПРОГРАМНОГО РІВНЯ.....	37
3.1 Розробка алгоритму	37
3.2 Конструювання програмного забезпечення	40
3.3 Архітектура програмного забезпечення	44
3.3.1 Стек технологій для виконання роботи	44
3.3.2 Операційна система FreeRTOS.....	45
3.3.3 Потoki FreeRTOS.....	45
3.3.4 Відмовостійкість та системи захисту.....	47
3.3.5 Годинник реального часу та NTP	51

3.3.6	Формат зберігання сценаріїв росту	52
3.3.7	Зберігання даних у EEPROM	54
3.4	Метод підключення до мережі.....	58
3.5	Висновки по розділу	67
4	ТЕХНІЧНА РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ АЛГОРИТМУ.....	68
4.1	Визначення переліку апаратного забезпечення	68
4.2	Компонування елементів.....	69
4.3	Прошивка МК	70
4.4	Перевірка прошивки	71
4.5	Тестування алгоритму	71
4.6	Висновки до розділу	75
5	РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ.....	77
5.1	Опис ідеї проекту.....	77
5.2	Технологічний аудит ідеї проекту	79
5.3	Аналіз ринкових можливостей запуску стартап-проекту.....	81
5.4	Розроблення ринкової стратегії проекту	91
5.5	Розроблення маркетингової програми стартап проекту	94
	ВИСНОВКИ	99
	ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	100

ПЕРЕЛІК ВИКОРИСТАНИХ СКОРОЧЕНЬ

IoT – Internet of Things

JSON – JavaScript Object Notation

ПЗ – програмне забезпечення

ОС – операційна система

HTTP – Hyper Text Transfer Protocol

API – Application Programming Interface

JS – JavaScript

CSS – Cascading Style Sheets

HTML – Hyper Text Markup Language

АЦП – Аналогово Цифровий Перетворювач

SPI – Serial Peripheral Interface

UART – Universal Asynchronous Receiver-Transmitter

TX – Transmit

RX – Receive

MAC – Media Access Control

TCP – Transmission Control Protocol

dBm – decibel-milliwatts

RFC – Request for Comments

EEPROM – Electrically Erasable Programmable Read-Only Memory

МК – Мікроконтролер

РН – pondus Hydrogenii

ЕС – Electrical Conductivity

БВ – бак відходів

МКСК – камера для замішування

ГРЧ – годинник реального часу

МФК – модуль флеш-карти

МЗ – модуль зв'язку

ПП – послідовний порт

IDE – Integrated Development Environment

CSV – Coma Separated View

ВСТУП

В даний час сучасні технології вже впроваджені практично в усі галузі людських інтересів. Тому можна зазначити, що більшість нововведень, які зараз впроваджуються, здебільшого, стосуються вдосконалення вже наявних рішень.

У сучасному світі можна виділити кілька проблем, пов'язаних з базовими потребами людства. Потреба в їжі є однією з головних. Такі фактори як збільшення популяції, скорочення площ родючих земель, пов'язаних з масовим вирубуванням лісів, глобальне потепління – все це негативно впливає на можливість вирощування достатньої кількості сільськогосподарської продукції. Одним з можливих варіантів розв'язання даної проблеми є впровадження нових технологій для оптимізації вирощування рослин і спрощення цього процесу.

В рамках даної магістерської роботи буде розроблено програмно-апаратний продукт, який буде спроможним вирішити існуючі проблеми і вдосконалити наявні рішення для кінцевого користувача. Розроблена система дозволить досягти оптимальної різниці між часовими, ресурсними витратами і якісними характеристиками одержуваного продукту.

Цифрова трансформація, яка зараз відбувається, спонукає організації до постійного оновлення та вдосконалення своїх бізнес-моделей. Значна частина цих змін пов'язана з впровадженням нових технологій. Лідери технологічних інновацій повинні прийняти новий спосіб мислення і нові методи, а також прийняти сам факт постійних змін. Стратегія, яка дозволяє організаціям заохочувати постійні інновації і інтегрує їх може досягти успіху у світі, який постійно змінюється. Технологічне майбутнє характеризуватиметься розумними пристроями, що забезпечують дедалі глибші цифрові послуги скрізь. Можна спостерігати як все частіше поєднуються цифровий та фізичний світ. Цифрові тенденції, поряд з можливостями, є рушієм для наступного покоління цифрового бізнесу та створення цифрових бізнес-екосистем. На сьогоднішній день будь-який успішний бізнес-проект характеризується використанням величезної кількості зв'язків між людьми, бізнесами та, звичайно, пристроями і послугами. Розроблюваний, у рамках цієї дисертації, пристрій також

має набір зв'язків між різними підсистемами. Це вимагає формування підходів для злагодженої роботи системи в цілому.

Автономні речі використовуються для автоматизації функцій, які раніше виконували люди. Їх автоматизація виходить за рамки автоматизації, передбаченої жорсткими моделями програмування, і вони використовують шляхи для забезпечення вдосконалення поведінки, яка більш природно взаємодіє з оточенням та людьми. Автономні речі бувають різних типів і діють у багатьох середовищах з різним рівнем здатності, координації та інтелекту.

Автономні речі часто мають форму фізичного пристрою, що працює в реальному світі. Прикладами можуть бути роботи, безпілотники та автономні транспортні засоби. Елементи IoT на основі штучного інтелекту, такі як промислові обладнання та побутова техніка, також є видом автономних речей. Автономні речі діють по спектру, від напіваавтономних до повноцінних автономних. Слово "автономний" використовується для опису таких речей, що можуть діяти без нагляду в обмеженому контексті або для виконання завдання. Автономні речі можуть мати різні рівні автономії. Наприклад, самонаправляючий пілосос може мати обмежену автономію, тоді як безпілотник може автономно ухилятися від перешкод. У міру поширення подібних речей очікується перехід від окремих інтелектуальних речей до сукупності декількох спільних. У цій моделі кілька пристроїв будуть працювати разом незалежно від людей або за участі людей. Наприклад, якщо безпілотник оглянув велике поле і виявив, що він готовий до збирання врожаю, він може відправити "автономний комбайн". Автоматизація приймає безліч різних форм. У даному випадку мова буде йти про управління внутрішнім середовищем і автоматизацію зрошувальної системи з добривами яка має назву фертигація.

Фертигація це сучасна агротехніка, яка забезпечує чудову можливість максимізувати врожайність та мінімізувати забруднення навколишнього середовища за рахунок підвищення ефективності використання добрив, мінімізації внесення добрив та збільшення рентабельності вкладеного добрива. Під час фертигації терміни, кількість та концентрація добрив легко контролюються. Великий асортимент добрив, як твердих, так і рідких, є придатним для фертигації залежно від

фізико-хімічних властивостей розчину добрива. Для великомасштабних польових операцій джерела твердих добрив, як правило, є менш дорогою альтернативою загальновживаним рідким складам. Розчинність цих добрив дійсно сильно відрізняється. При переході на тверде джерело добрив можна уникнути проблем у резервуарах, забезпечивши додавання достатньої кількості води в розчин. Тип добрива для фертигації повинен бути високоякісним, з високою розчинністю та чистотою, що містить низький рівень солі, з прийнятним рН, і воно повинно входити в програму управління. Автоматизовані системи фертигації дозволяють виробникові зрошувати й удобрювати великі площі сільськогосподарських культур з мінімальними виробничими відходами і персоналом. Кінцева мета автоматизації це полегшення роботи людині. Це зумовлює створення системи клімат-контролю, щоб контролювати температуру внутрішньої середи, забезпечення отримання врожаю оптимальної кількості поживних речовин на постійній основі, налаштування локальної зміни дня і ночі у рамках системи, надання можливості віддаленого моніторингу.

Актуальність: наразі немає системи для користувача, який не володіє спеціальними технічними навичками, для автоматичного зрошування в ґрунті. Для користування існуючими пристроями користувач повинен бути професіоналом та вміти налаштовувати кожен окремий модуль, а саме регулятор кислотності, систему автоматичної подачі добрив, систему підтримки температури та змусити працювати це все спільно. Також необхідно налаштовувати роботу за графіком та синхронізувати всі модулі, що є трудомістким та має велику вірогідність допущення помилок. Проєкт, який розроблюється у рамках магістерської дисертації, вирішує ці проблеми, об'єднуючи наявність обладнання для повного циклу зрошування у одній системі, наявність дистанційного керування та простоту підключення та використання.

Метою є розробка системи, що забезпечить можливість покращеного зрошення рослин шляхом підтримки ідеальних параметрів середовища з мінімальним втручанням людини.

Об'єкт дослідження: автоматизація процесу зрошення рослин.

Предмет дослідження: система автоматизації процесу зрошення рослин.

Завдання роботи:

- розробка моделі системи;
- розробка структурної та функціональної реалізації системи;
- програмування мікроконтролера;
- компонування відокремлених модулів у єдину систему;
- прошивка мікроконтролера;
- тестування системи.

Матеріалом для дослідження є системи «IoT», системи автоматизації теплиць, гідропонік та «гроубоксів», ферм, крапельних поливів.

1 ОГЛЯД ТЕХНОЛОГІЙ І СИСТЕМ АВТОМАТИЗАЦІЇ

1.1 Змістовний опис і аналіз предметної області

Процес вирощування рослин дуже трудомісткий. Він потребує уважного догляду та контролювання наступного переліку завдань:

- стежити за рівнем вологості ґрунту і поливати рослину;
- контролювати і коригувати рівень Ph в розчині для поливу;
- здійснювати аналіз ґрунту на мінералізацію;
- додавати добрива в розчин для поливу;
- стежити за температурним режимом;
- стежити за циклами роботи освітлення;
- стежити і коригувати рівень CO₂;
- стежити і коригувати вологість повітря;
- стежити за циклами життя рослини і змінами норм утримання та харчування.

На цьому етапі треба зупинись на найскладнішому завданні, а саме зрошування оптимальним розчином. Існує декілька підходів організації подачі поживних речовин:

- не автоматизований;
- часткова автоматизація;
- повна промислова автоматизація;
- повна користувацька автоматизація.

Перший, з перелічених, спосіб передбачає виконання всіх дій самостійно.

Необхідно виконати наступні кроки:

- виміряти необхідний об'єм води;
- скорегувати рН цієї рідини;
- виміряти необхідну кількість добрив;
- знайти інформацію про те, як користуватись цими добривами правильно;
- постійно слідкувати за вологістю ґрунту;
- слідкувати за температурою повітря.

Недоліками даного підходу є не оптимальне використання часу, велика трудомісткість, відсутність можливості віддаленого моніторингу, неефективне використання добрив, збільшення можливості помилки внаслідок людського фактору.

Наступний підхід, а саме часткова автоматизація, передбачає такі завдання:

- пошук і придбання модулю підтримки рН;
- пошук і придбання модулю автополиву;
- автоматизація освітлення за допомоги розумної розетки;
- автоматизація температури за допомоги модулю підтримки температури.

Недоліками такого підходу є складність збірки та відладки для того, щоб всі модулі спільно працювали коректно, відсутність системи моніторингу, відсутність віддаленого управління. Та можливо найголовніший недолік полягає в тому, що таку систему неможливо налаштувати таким чином, щоб вона змінювала свої параметри динамічно, відповідно до часу та періоду росту рослини, а кожному виду рослин на кожному етапі необхідні різні кількісні показники. Єдиним рішенням цієї проблеми є постійне переналаштування вручну.

Повна промислова автоматизація використовується на великих підприємствах, які виробляють продукцію у величезних масштабах. Цей підхід передбачає дуже великі матеріальні витрати. Також з'являється питання про можливість зміни параметрів у необхідний час на певному етапі росту рослин.

Повна користувацька автоматизація, саме цей підхід реалізований у розроблюваній системі, не потребує від користувача специфічних знань про те, як треба ростити обрану рослину, достатньо вибрати її у бібліотеці.

Пристрій має підтримку сценаріїв. Користувач обирає сценарій, після чого пристрій буде змінювати підтримувані параметри протягом усього росту рослин. За необхідності є підтримка створення особистих сценаріїв, які можна завантажити у пристрій. Така автоматизація удосконалює функціонал часткової автоматизації, додаючи роботу за графіком. Також така система має повну автономність, окрім наповнення баків з водою та добривами і посадки і збору врожаю.

Наразі існує декілька варіантів як вирощувати рослину. Для цього можна використовувати ґрунт або гідропоніку. Можна казати про те, що функціонал системи не буде залежати від того, який варіант буде обрано, проте від цього залежить як вона буде побудована. Для того, щоб підібрати оптимальний варіант для конкретної системи, необхідно розуміти, чим відрізняються процеси обох.

Гідропоніка представляє собою систему, яка дозволяє рослинам рости без ґрунту.



Рисунок 1.1 – Система гідропоніки[1]

Замість того, щоб покладатися на ґрунт для отримання поживних речовин, рослини, вирощені в гідропоні, отримують мінеральні поживні речовини безпосередньо, а вода доставляється безпосередньо через кореневу систему. Поживні речовини для гідропоніки отримують з мінеральних солей, які по суті такі ж, як і в ґрунті, тільки більш доступні.

На момент придбання, ґрунт представляє собою суміш органічних матеріалів з додаванням мінералів. У більшості випадків мінерали в ґрунті, який був придбаний у магазині, мають однакову природу отримання, що і мінерали для гідропоніки. Різниця в тому, що процес, який відбувається під час живлення рослини, представляє собою повільне вивільнення, тоді як гідропонні мінерали вивільняються швидко з швидким

поглинанням, що забезпечує оптимальні результати і більш швидке зростання. У ґрунті коріння рослин повинні шукати поживні речовини. Ось чому зазвичай ґрунтову рослина має набагато більшу кореневу систему, ніж гідропонних. Завдяки гідропонії поживні речовини доставляються прямо в кореневу систему майже в хірургічних кількостях. Цей метод гарантує, що рослини отримують точно потрібну кількість поживних речовин в потрібний час, дозволяючи рослині витратити свою енергію на виробництво листя, стебел, листя і плодів, замість великих корневих систем. Термін гідропоніка зараз дуже часто використовується для всіх середовищ, крім ґрунту. Операції з вирощування на водній основі без стаціонарної середовища називаються «культивуванням в розчині». Тому у даному випадку під терміном гідропоніка будуть визначатись методи, при яких корені рослини знаходяться в постійному контакті з водним розчином. Потім в воду додають поживні речовини в рідкій формі, створюючи новий розчин. Він буде складатися тільки з найнеобхіднішого для рослини і дасть садівнику набагато більше контролю над кінцевим результатом.

Ґрунт часто називають зеленою частиною «блакитної планети». І хоча він домінує лише на 30% поверхні Землі, саме тут росте більшість рослин, і це відбувається вже мільйони років. Через це в ґрунті скупчилися мінерали і органічні речовини, які дуже важко відтворити будь-яким іншим методом. Ось чому багато садівників будуть вирощувати продукцію на відкритому повітрі в ґрунті. Але все ж важливо відзначити, що більшість ґрунтознавців додають поживні розчини або багаті поживними речовинами матеріали, такі як відливання дощових черв'яків і гній, для поліпшення свого середовища. При вирощуванні на вулиці спостерігатимуться перепади температури, нестримний вітер і навіть коливання вологості. Їх важко передбачити і неможливо освоїти. Можна лише пристосуватися до зовнішнього середовища і сподіватися на краще. Але ця проблема коригується за умови вирощування у замкненому просторі. Непідготовлений ґрунт також містить органічні речовини та бактерії, які можуть бути не надто сприятливими для здоров'я вашої рослини. Їх буде важко визначити, поки на рослині не проявляться візуальні зміни. Будь то у формі мінерального порошку або розчиненого у воді, продукти з

макроелементами матимуть три основні елементи: азот, фосфор та калій. Ці поживні речовини узагальнено співвідношенням N-P-K[2], що складається з трьох цифр на передній частині пляшки з поживними речовинами. Кожне значення представляє об'ємний відсоток відповідної поживної речовини в розчині. У ґрунтове середовище можна додавати гній, який є багатим поживними речовинами матеріалом. Це те, що гідропонний розчин не може повторити. Ґрунт наповнений мікробами, які допомагають перетворити органічний матеріал, такий як відливи гуано та глистів, у корисні поживні речовини для рослини. У свою чергу гідропоніка вимагає насичення рослини повною кількістю мікро- та макроелементів. Незалежно від якості вихідного ґрунту, ймовірно, знадобляться додаткові добавки, щоб отримати кращі результати. Саме тут гідропонна система переважно відрізняється від ґрунтового середовища. Такі мікроелементи, як залізо, мідь та магній, широко доступні в більшості ґрунтових середовищ. Отже, гідророзчини повинні містити більше таких для компенсації. Вони також потребують більше азоту, макроелементів, яких багато в ґрунті, але не так багато у джерелах води. Саме це зумовлює більший відсоток азоту у гідрогенних поживних речовинах на вегетативній стадії. Також, ґрунт зберігає природну сутність рослини. Як приклад можна навести те, що деякі люди віддають перевагу зовнішньому аромату ґрунту, який він надає квітам. Також ґрунт є кращим варіантом, коли рослини вирощуються не постійно, тобто періодично. Ґрунт вимагатиме набагато менше уваги, оскільки більшу частину роботи він виконуватиме самостійно.

У даному проєкті буде використано ґрунт. Це зумовлено тим, що наразі системи зазвичай базуються на гідропонних системах. Ниша зрошування землі відносно вільна. Також з урахуванням того, що кокосовий субстрат має нейтральний рН та мінералізацію, що дозволяє наповнити його тільки необхідними мінералами та задати необхідний рН, буде обрано підхід з використанням кокосового субстрату як основу до вирощування.

Вода, яка буде використана для зрошування ґрунту, подається у ємність, в яку, за допомоги перистальтичних насосів, буде додано необхідну кількість добрив та рН коректорів.

Автоматизація такої системи може мати наступні переваги:

- дозволяє значно скоротити або повністю виключити трудовитрати і можливість людської помилки при змішуванні сумішей;
- використання води і добрив можна звести до мінімуму, так як ці системи можуть легше пристосуватися до повторного використання води;
- економить час, так як схожі рослини можна згрупувати і поливати в одній або декількох зонах;
- датчики можуть відслідковувати, скільки води і добрив, в яких потребують рослини, вони отримують;
- рівномірні, послідовні цикли поливу для отримання більш однорідного урожаю з кращими кількісними результатами;
- надзвичайна окупність інвестицій, вона може бути досягнута для тепличних проєктів будь-якого розміру, але в цілому, чим більше проєкт, тим економічно доцільніше.

1.2 Аналіз існуючих рішень

Варто зазначити, що на даний момент немає прямих аналогів розроблюваної системи. Майже всі, запропоновані на ринку пропозиції, використовують систему гідропоніки, на відміну від розроблюваної системи. Далі буде розглянуто успішні проєкти, які вирішують завдання оптимізації системи для вирощування рослин, та надано порівняння функціональних характеристик аналогів з розробленою системою.

Одним з успішних прикладів у даній галузі є Grobo, зображена на рисунку 1.2.



Рисунок 1.2 – Grobo [3]

Grobo це найпростіший продукт, який пропонує клієнтам можливість вирощувати свіжі фрукти, зелень та овочі у власному будинку з мінімальними зусиллями. Як зазначають виробники, Grobo поєднує технології вирощування, використовуючи датчики для контролю стану рослини, автоматично регулюючи її графік поливу, а потім повідомляє, коли воно готове до збору врожаю. Він підходить для ряду культур – включаючи помідори, полуницю, перець, зелень, капусту тощо. Grobo надає додаток, який за бажанням користувача, надає підказки на різних фазах росту рослини для покращення процесу. Grobo використовує наступні датчики, які постійно збирають та передають дані щодо середовища вирощування рослини:

- ЕС сенсор;
- Grobo.pH;
- сенсор для вимірювання рівня рідини.
- сенсор освітлення
- сенсор температури

ЕС означає електропровідність. Цей датчик використовується для вимірювання рівня поживних речовин і твердих речовин у воді. Далі дані використовуються для регулювання рівня дозування поживних речовин. Grobo.pH відповідає за підтримання постійного рівня pH в межах оптимального діапазону. Три датчики рівня води у резервуарі використовуються для вимірювання рівня води. На різних етапах циклу росту рослини рівень води потрібно регулювати для оптимального зростання. Ці датчики забезпечують, щоб резервуар для води щоразу наповнювався до потрібної точки. Датчики використовуються на рахунок таких, які потенційно можуть бути використані і у системі, що розробляється.

Планується замінити датчики поплавки на більш точний датчик ваги, але не відмовлятися від них. Вони потенційно будуть потрібні для калібрування тензо-датчиків вагів.

Одним з недоліків даної системи є обмежений розмір, який накладає обмеження на обсяг продукції. Більш детально функціональні обмеження розглянуті у таблиці нижче.

Ще одним прикладом аналогічної системи є Leaf.

More about LEAF



Рисунок 1.3 – Leaf[4]

Виробник зазначає, що система покриває ряд функцій, таких як точний контроль температури та вологості, автоматичне дозування поживних речовин, необов'язкове підключення для автоматичної заміни води. Система також реагуватиме на вимірювання, тобто, якщо вона відчує, що у рослин недостатньо поживних речовин, вона може дозувати більше, або якщо їй потрібно більше світла, вона може автоматично покращити навколишнє середовище. Користувачі можуть також керувати цими змінними вручну. Також присутнє автоматичне зрошення за допомогою вибору типу рослини. Після цього система починає змінювати параметри до заданих.

Досить велика ціна цієї системи є одним з недоліків. Також недоліком можна вважати не описаний алгоритм проходження процесу зрошення що може викликати непередбачений результат для користувача.

Наступна система, яка буде розглянута, має назву BC North L, зображена на рисунку 1.4.



Рисунок 1.4 – BC North L[6]

Система заздалегідь запрограмована із синхронізацією часу, поливом, впорскуванням CO₂, циркуляцією повітря та насосами. Єдиним з доступних можливих варіантів керування є керування за допомогою панелі управління. Система здатна відстежувати стан здоров'я рослини, рівень рН або рівень поживних речовин.

Недоліком даної системи є відсутність контролю та повідомлення про необхідність заповнення запасів води, також відсутній контроль вологості ґрунту, автоматизація поливу.

З наступним функціоналом система частково перестає конкурувати з розробленою через відсутню автоматизовану зміну заданих параметрів з плином часу. Такий підхід має відсутність зручності через перекладання обов'язків зміни параметрів у часі на користувача. Відсутність системи сповіщень може призвести до зупинки та негативному впливі на рослину через відсутність оновленої води у резервуарі системи.

Далі представлена таблиця, у якій порівняно функціональні характеристики вищезгаданих систем та деяких інших рішень з розроблюваної у рамках магістерської дисертації системою.

Функціональність:	Власна система	Seedolab	Grobo	Leaf	BC North L	Cloud Ponics	Mary	Harvstbox
Контроль вологості ґрунту, автоматизація поливу								
Контроль і корекція Ph в суміші для поливу								
Контроль і корекція мінералізації ґрунту (EC)								
Контроль і повідомлення про необхідність поповнення запасів води і добрив								
Управление сторонним устройством освещения								
Контроль температури повітря								
Управління стороннім пристроєм вентиляції								
Управління пристроями нагріву / охолодження повітря								
Онлайн відео контроль								
Контроль і управління процесом вирощування за допомогою web інтерфейсу і mob apk								
Система безпеки і інформування								
Можливість підключення до пристрою без доступу до мережі Internet, через Bluetooth								
Отримання звітів про процес вирощування (витрати води, добрив, кількість отриманого світла, швидкість росту)								

Рисунок 1.5 – Огляд існуючих рішень

1.3 Можливість використання у аграрній промисловості

Теплиця забезпечує контрольоване середовище з урахуванням рослинних потреб, що вирощуються всередині. Традиційно мікроклімат та агрономічні параметри налаштовуються вручну та непослідовно. Дослідження та розробка бездротових сенсорних мереж у теплицях почалися на початку 2000-х рр. Тоді почали застосовувати технологію бездротового зв'язку для моніторингу, надсилання повідомлень раннього попередження та дистанційного керування за допомогою спрощених правил. Компактні розміри, надійність та економічна ефективність модулів, а також гнучкість для розробки нестандартних додатків, крім простої установки, змусили цю технологію завоювати увагу та популярність для моніторингу та контролю у відкритому полі та у закритих приміщеннях. Завдяки цим дослідженням, у 2015 році була реалізована локально-глобальна стратегія управління, що підтримує синхронізацію онлайн та оффлайн конфігурації системи, спрацьовування та управління в автономному режимі, щоб реагувати на нестабільне мережеве з'єднання у сільській місцевості. Була перевірена функціональність системи протягом 10-денного періоду збору даних для автоматизованого моніторингу вмісту вологи в ґрунті та контролю за заплідненням при вирощуванні томатів. Їх результати показали, що за нестабільної мережі система мала лише похибку 0,78% і

забезпечувала вимірювання вмісту вологи в ґрунті 99,2%, що показує потенціал для довгострокового моніторингу мікроклімату в теплицях.

Існує обмеження щодо того, що можна виміряти, і сільськогосподарські практики виконуються за заздалегідь визначеним графіком. З іншого боку, погода змінюється протягом дня, а «невидимі» умови, такі як відчинені двері або зараження на ранніх стадіях, постійно впливають на тепличне середовище та загрожують шкодою посівам. Розумні теплиці об'єднують звичайні сільськогосподарські системи та нові технології IoT для повної видимості та автоматизації.



Рисунок 1.6 – Теплиця з використанням IoT[5]

Оснащені сучасними сенсорними та комунікаційними технологіями, розумні теплиці автоматично збирають та забезпечують цілодобову інформацію про навколишнє середовище та урожай. Зібрані дані подаються на платформу IoT, де аналітичні алгоритми виявляють вузькі місця та відхилення. Відповідно, операції з опалення та вентиляції, поряд із зрошенням та обприскуванням, можуть регулюватися на вимогу. Постійний моніторинг даних сприяє розробці прогнозних моделей для оцінки ризиків захворювань сільськогосподарських культур та

зараження. Надаючи значну інформацію про урожай, розумна теплиця дозволяє виробникам мінімізувати робочу силу, підвищити ефективність використання ресурсів та хімічних речовин, одночасно оптимізуючі показники врожайності. Датчики IoT дозволяють фермерам збирати різні точки даних з безпрецедентною деталізацією. Вони надають інформацію в режимі реального часу про критичні кліматичні фактори, включаючи температуру, вологість, вплив світла та діоксид вуглецю в теплиці. Ці дані вимагають відповідних налаштувань, щоб підтримувати найкращі умови для росту рослин, одночасно забезпечуючи енергоефективність. Паралельно датчики руху та прискорення допомагають ідентифікувати двері, які ненавмисно залишаються відкритими, щоб забезпечити суворо контрольоване середовище. Окрім параметрів навколишнього середовища, розумні теплиці дозволяють фермерам встигати вчасно проводити необхідні операції над рослинами. Це забезпечує зрошення та підживлення добрив на відповідному рівні для отримання максимальної кількості кращого за якістю врожаю. Наприклад, показники об'ємного вмісту води в ґрунті вказують на те, чи не знаходиться рослина в надлишку рідини. Аналогічним чином, вимірювання солоності ґрунту дають корисну інформацію про вимоги для добрив. На основі цих даних, спринклерні та оприскувальні системи можна автоматично вмикати для задоволення потреб врожаю в реальному часі, мінімізуючи ручне втручання.

Теплиці з високою цінними культурами є вразливою мішенню для злодіїв. Оскільки традиційні мережі спостереження із системами відеоспостереження дорогі у впровадженні, багато виробників не мають ефективної системи безпеки. У цьому контексті датчики IoT у розумних теплицях забезпечують доступну інфраструктуру для контролю стану дверей та виявлення підозрілих дій. Пов'язані з автоматизованою системою сигналізації, вони миттєво повідомляють виробників, коли виникає проблема безпеки.

Розкидані на великих географічних територіях, парникові комплекси комерційного рівня вимагають бездротового зв'язку на великі діапазони з потужними можливостями проникнення. Окрім забезпечення надійної передачі даних, таке рішення дозволяє встановлювати шлюз ближче до джерела живлення, щоб зменшити

траншею траншеї. Масштабованість – це ще один фактор, який слід враховувати, оскільки він мінімізує кількість шлюзів, необхідних для економії на апаратних витратах, витратах на встановлення та управління. Перш за все, зв'язок повинен бути високоефективним, щоб датчики могли знаходитись роками з мінімальним обслуговуванням.

Важливим фактором при розробці ефективної WSN для теплиці є кількість вузлів, їх розташування, відстань між вузлами, антена та робоча частота на основі стану мікроклімату теплиці. Дослідження показали, що на поширення радіохвиль сильно впливає сильне парникове середовище.

Звичайні алгоритми контролю мікроклімату розроблені на основі серії опорних значень, відомих як задані значення. Потім контролер регулює виходи за допомогою впроваджених правил для досягнення стабільності шляхом мінімізації помилок між посиланнями та входами. Контроль мікроклімату в теплиці є завданням високого рівня через кількість залучених змінних, які зв'язані та взаємопов'язані, утворюючи складну нелінійну систему.

Ідентифікація та моделювання різних параметрів у теплицях були темами численних дослідницьких робіт, спрямованих на підвищення ефективності виробництва (тобто підвищення врожайності та якості, максимальну віддачу) шляхом оцінки та регулювання параметрів мікроклімату та моделювання мікрометеорології. Рівень підходів варіюється від простих моделей та керування зворотним зв'язком на основі таймера до більш вдосконалених рішень, таких як безмодельна стратегія управління, нелінійні методи управління, адаптивне управління та система адаптивного управління, надійний, оптимальний контроль, моделі енергетичного балансу та прогнозний контроль на основі моделей. Важливим недоліком використання передових методів управління є складність у розробці динамічної моделі для імітації поведінки змінних. Зрештою, метою будь-якої з цих систем управління є мінімізація витрат на одиницю продукції та збільшення віддачі за рахунок досягнення високої врожайності та якості. Управління виробництвом у теплиці вимагає прийняття рішень за кількома завданнями та часовими шкалами. Ці рішення в основному пов'язані з управлінням умовами росту врожаю, періодом

зрощування та такою практикою, як виробництво розсади, зрошення на основі подій та контроль та управління природоохоронними моделями росту.

З прогнозованою зростаючою ринковою вартістю, автоматизовані теплиці, без сумніву, є головною тенденцією ведення сільського господарства[7]. Така система поєднує звичайні сільськогосподарські системи та нові технології IoT для повної видимості та автоматизації. Це допомагає визначити неефективність та боротися з проблемами, які заважають сільськогосподарській праці, з метою захисту врожаю та максимізації врожаю.

1.4 Висновки до розділу

Обрано для розробки систему для підтримки життєдіяльності рослини у ґрунті через відносно незайняту нішу.

Щоб бути конкурентоспроможною, система має бути з розряду IoT, тому у системі обов'язково має бути присутнім МЗ, який має змогу підключитися до мережі Інтернет.

Отримано перелік параметрів, зміну та підтримку яких система має автоматизувати. Отримано первинну уяву про взаємодію елементів системи з можливими елементами, зображених у додатку 1. Ці дані будуть використані для реалізації технології на структурному та функціональному рівнях. Також спроектовано діаграму прецедентів зображену у додатку Ж.

На початковій стадії сплановано роботу системи у середовищі спеціалізованого ящика для вирощування через легше налаштування у малих розмірах. У подальшому можливе масштабування системи на будь-яку кількість рослин та для використання у теплицях будь-якого розміру. Масштабування відбувається за рахунок збільшення потужності pomp, підбору відповідного джерела живлення, збільшення величин резервуарів, встановлення у конфігурації системи цих величин.

2 РЕАЛІЗАЦІЯ НА СТРУКТУРНОМУ ТА ФУНКЦІОНАЛЬНОМУ РІВНІ

2.1 Структура системи

Для побудови системи необхідно розробити основні модулі системи, такі як: керуючий орган, виконавчий орган, регулюючий орган. Кожен з них виконуватиме свою функцію та буде взаємодіяти один з одним для досягнення мети – автоматизації процесу зрощення. Структурна схема зображена у додатку А.

2.1.1 Керуючий орган

Керування процесом здійснює мікроконтролер Mega2560. Цей мікроконтролер обрано через наявну кількість GPIO у розмірі 54 одиниць, 12 з яких підключені до АЦП, 4 з яких є апаратними послідовними портами, 2 з яких є апаратною шиною SPI та ще 2 шиною I2C. Цього оснащення мікроконтролера достатньо, щоб підключити датчики, виконавчий орган та периферію, а саме ГРЧ, МФК та МЗ.

Щоб керуючий орган, зображений на рисунку 2.1, міг позиціонувати себе у часі для слідування сценарію з точністю до хвилини, у керуючому органі присутній ГРЧ. ГРЧ є незалежним модулем, та продовжує роботу деякий, встановлений умовами енергоспоживання та ємності акумуляторної батареї, час, щоб у умовах тимчасової відсутності мережі, надати змогу керуючому органу продовжити роботу з правильної точки.

Для зберігання сценарію процесу зрощення наявний МФК. МФК являє собою сховище даних, яке може отримати запит від МК на отримання даних.

Для керування регулюючим органом потрібно реле – виконавчий орган. Його було включено у керуючий орган для цілісності системи.

Наявний МЗ забезпечує обмін даними з системою керування. Передбачається використання технології Wi-Fi. Модуль зв'язку виступатиме у ролі WebSocket-клієнту та робитиме обмін даними у режимі реального часу. Це необхідно для мінімізування часу відгуку системи, тому що це система реального часу. Дані передаватимуться у форматі JSON.

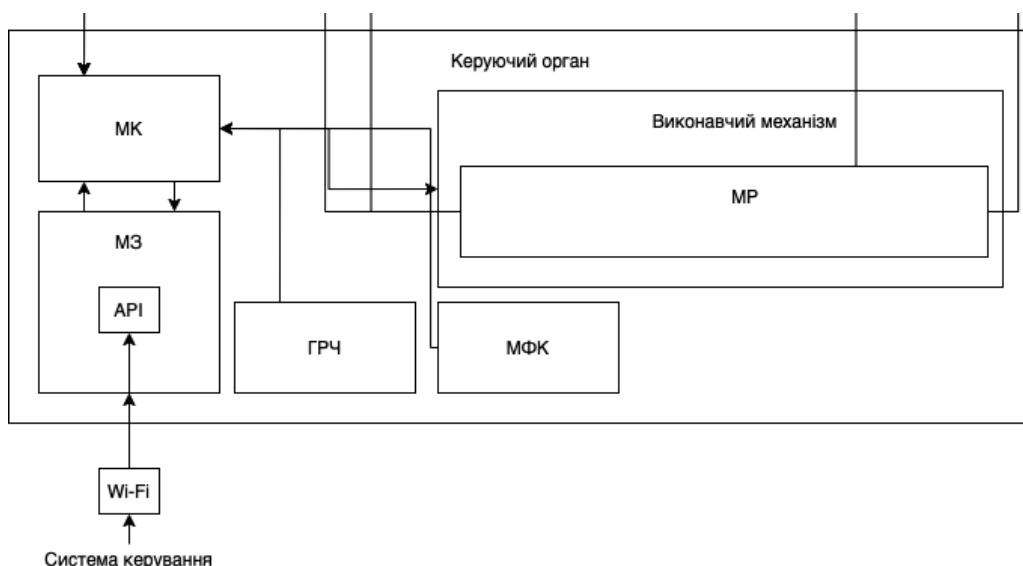


Рисунок 2.1 – Схема керуючого органу

2.1.2 Регулюючий орган

Регулюючий орган, зображений на рисунку 2.2, потрібен для врегулювання процесів у МКСК. Для врегулювання процесів МКСК регулюючий орган працює із рідинами. Рідина переміщується за допомогою pomp. Зворотній зв'язок виконується за допомогою датчиків наявних у МКСК.

Існує два типи pomp у даній системі: мембранна pompa із продуктивністю 2 л/хв. , та перистальтична, з продуктивністю 100мл/хв. Це зумовлено об'ємами, які потрібно прокачати кожній pomпі. Мембранні помпи перекачують велику кількість рідини, а саме: чисту воду, готовий розчин, дренаж, відходи. Перистальтичні помпи покликані перекачувати добрива – невеликі кількості добавок, що зазвичай сягають близько 2мл на 1л води.

Для виконання зворотнього зв'язку МКСК включає у себе датчики: датчик температури, датчик кислотності, датчик електропровідності, датчик рівня рідини. Це дозволяє керуючому органу припинити подачу добрив у потрібний момент. Керуючий орган має датчики, що потребують технічного обслуговування, а саме: калібрування датчиків, очищення поверхні, контроль цілісності системи. Ці операції проводить користувач самостійно. Підсистему контролю цілісності системи, а саме патрубків, герметичності МКСК буде розроблено у розділі 3.

2.2 Функції елементів системи

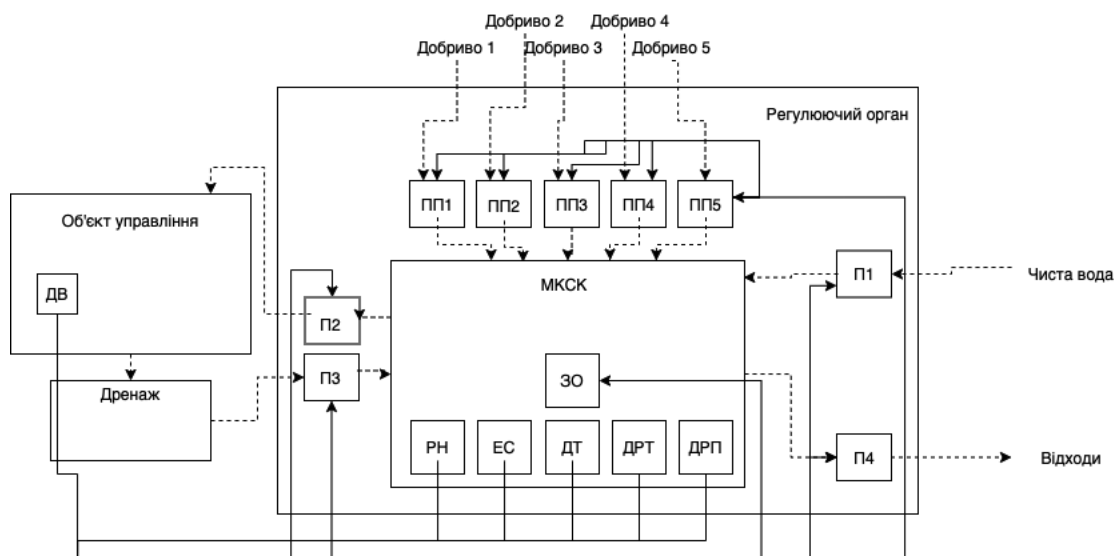


Рисунок 2.2 – Регулюючий орган

Керуючий орган, зображений на рисунку 2.3, виконує функцію управління виконавчим та регулюючим органом. Він має також виконувати функцію спостереження за процесом, інформуючи систему управління збираючи дані з датчиків. У ролі виконавчого органу виступає модуль реле, який під керівництвом керуючого органу подає сигнали регулюючому органу, зображеному на рисунку 2.2. Регулюючий орган, за допомогою зворотнього зв'язку має врегулювати стан об'єкта керування.

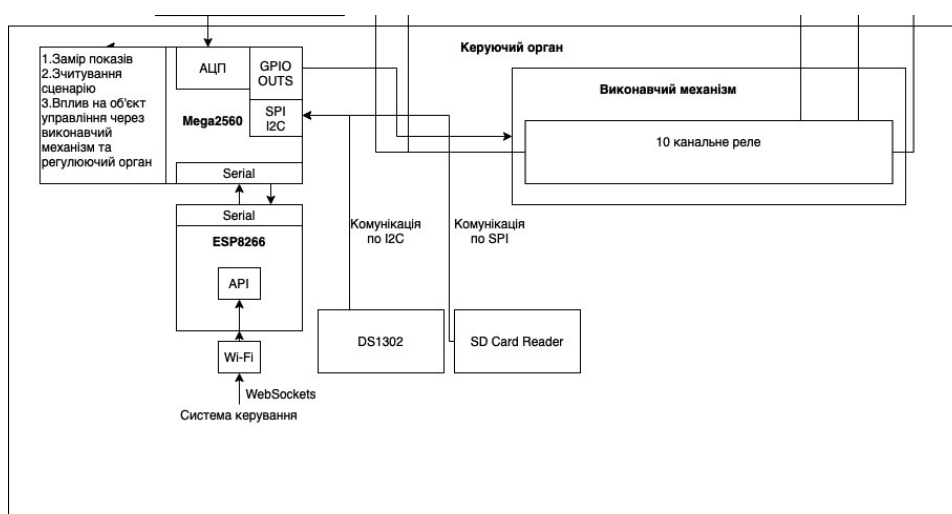


Рисунок 2.3 – Функціональна схема керуючого органу

2.2.1 Функції мікроконтролера

МК має зчитувати дані з датчиків, посилати сигнали для виконавчого органу, та керувати процесом. Для перетворення аналогового сигналу від датчиків, МК використовує вбудований 8 бітний АЦП. Прецедентом для відтворення алгоритму поливу є вологість, заміряна за допомогою датчика вологості, нижча ніж зчитана з периферійного пристрою – МФК. Функціональна схема показує зв'язок елементів системи.

2.2.2 Функції МЗ

Для пересилання сценарію від системи керування у систему вбудований МЗ. МЗ за допомогою технології WebSockets, перетворює отримані інструкції процесу зрошення, передає на МК за допомогою ПП, та МК зберігає їх у МФК.

МК має запрограмоване переривання за таймером кожні 5000 мс. Воно передає дані з датчиків у МЗ, який у свою чергу відсилає їх у форматі JSON до системи управління.

2.2.3 Функції МФК та ГРЧ

За допомогою ГРЧ, керуючий орган, зображений на рисунку 2.3, визначає точну дату та час. Щоб встановити граничні параметри стану системи МК порівнює дату і час, отримані від ГРЧ, та границі періоду сценарію. Якщо «зараз» є у границях періоду сценарію зрошення, то цей етап завантажується у пам'ять, та є основою для алгоритму поливу.

2.2.4 Функції МКСК

Щоб проводити операції із рідинами потрібна камера, у яку будуть підведені патрубки та датчики. МКСК є складовою регулюючого органу, зображеного на

рисунку 2.4, оскільки безпосередньо у ній відбуваються усі фізичні процеси, якими керує алгоритм, описаний у розділі 3.

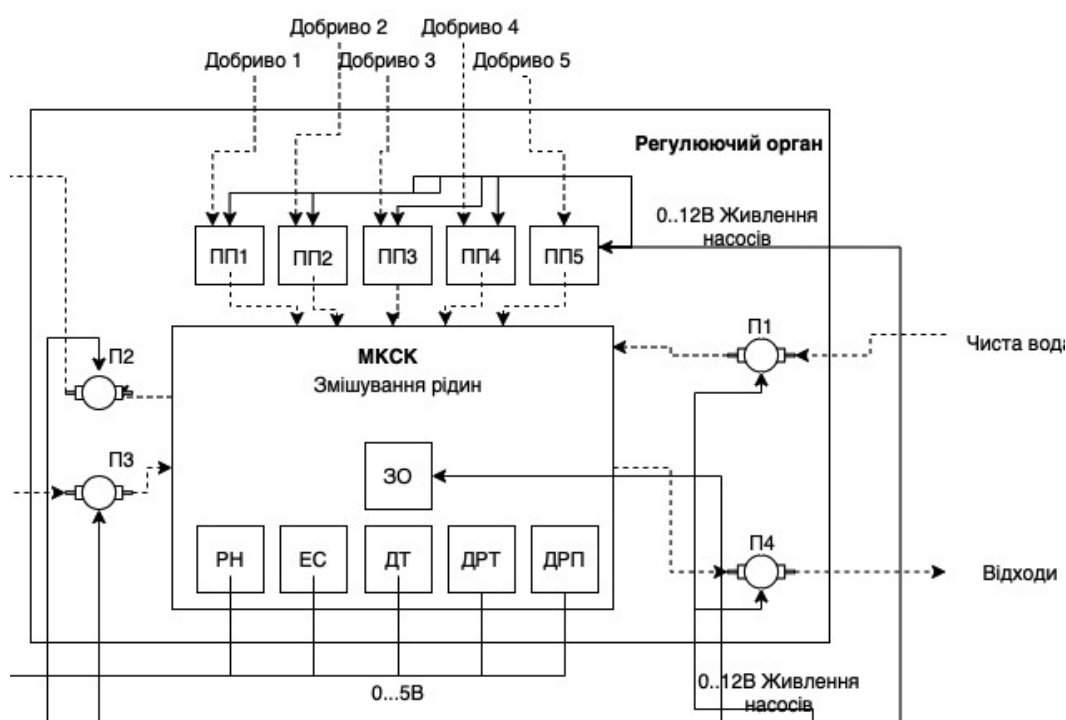


Рисунок 2.4 – Регулюючий орган

Неможливо провести зрошення, уникнувши проходження рідини через камеру. Щоб створити розчин потрібно пройти алгоритм, наприклад, у спрощеному варіанті: закачати чисту воду у камеру, заміряти кислотність, додати регулятор кислотності для встановлення у норму, зазначену у МФК, заміряти електропровідність, додати добрив для встановлення у норму, зазначену у МФК, провести зрошення за допомогою помпи номер 2.

2.3 Взаємодія елементів електронної схеми

Логічний рівень у плати ATmega2560 дорівнює 5В. Периферія, така як: датчики вологості, модуль карти пам'яті, модуль годинника реального часу, драйвери двигунів, реле – має відповідний логічний рівень. Це означає що їх можна підключати для плати, та логічні рівні будуть співпадати.

Wi-Fi модуль ESP8266 має живлення 3.3В, та логічна одиниця для цієї плати сягає 3.3В. Експериментально було встановлено, що плата ESP8266 може працювати з логічним рівнем 5В, але надійність такого процесу не може бути гарантована. Для того, щоб плати могли спілкуватися використовуючи UART, треба привести логічні рівні до однакового значення. Для цього можна використати двосторонній конвертер логічних рівнів, зображений на рисунку 2.5. Наприклад Sparkfun Bi-Directional Logic Level Converter.

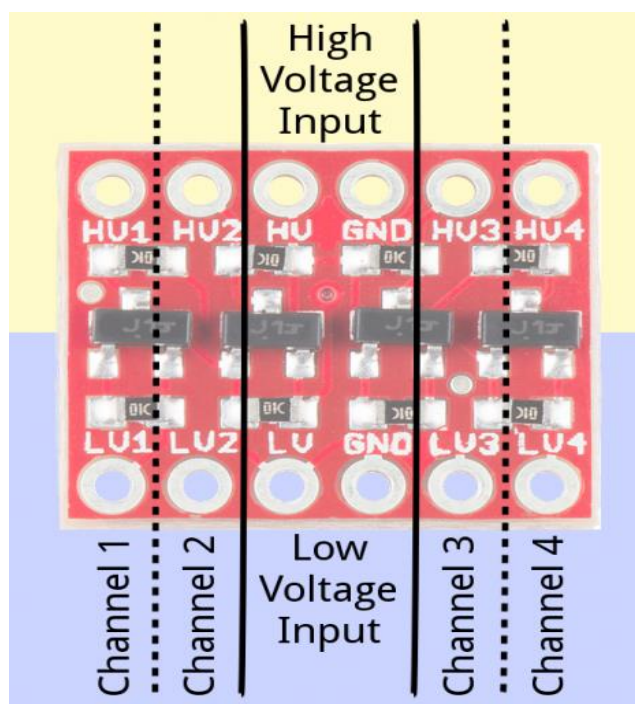


Рисунок 2.5 – Опис виводів конвертора[8]

Для того щоб скористатися конвертером потрібно під'єднати високу опірну напругу до частини High Voltage Input та низьку до Low Voltage Input. На платі присутні 4 канали даного призначення. Як заявляє розробник усі вони працюють незалежно один від одного та мають працювати у обох напрямках. Після цього можна використовувати будь-який з 4 доступних каналів для перетворення логічних рівнів. Плата ATmega2560 підключена до “високої”, з напругою 5В, частини, плата ESP8266 до “низької”, з напругою 3.3В.

2.4 Режим роботи ESP8266 та Mega2560

Вай-фай модуль, у момент підключення до мережі та серверу, починає прослуховувати повідомлення через послідовний порт від головного мікроконтролера системи. Це повідомлення має формат JSON.

Нотація об'єктів JavaScript – це відкритий стандартний формат файлу і формат обміну даними, в якому використовується створений людиною текст для зберігання і передачі об'єктів даних, що складаються з пар атрибут-значення і типи даних масиву. Це дуже поширений формат даних з широким спектром додатків, наприклад, він служить заміною XML в системах AJAX.[9]

JSON – це формат даних, що не залежить від мови. Він був отриманий з JavaScript, але багато сучасних мов програмування включають код для генерації та аналізу даних в форматі JSON. Офіційний тип інтернет-носія для JSON – application / json. Імена файлів JSON використовують розширення .json.

Стандарти ECMA і ISO описують тільки дозволений синтаксис, тоді як RFC охоплює деякі аспекти безпеки і взаємодії.

Для взаємодії між сервером та пристроєм, а також інтерфейсом керування, необхідно розробити документацію API. Для систем, які використовують Websockets, є: назва обробника, модель даних що можна передати обробнику, модель даних що можна отримати від обробника.

Для підвищеної безпеки для зв'язку буде використано протокол SSL та придбано SSL-сертифікат[10].

2.5 Висновки до розділу

У розділі визначено основні елементи органів системи. Побудована структура на основі якої побудовано функціональну схему, зображену у додатку Б.

Описано взаємодію кожного елемента для кожного органу системи, їх призначення, методи взаємодії. Отримано уяву про органи системи, що полегшить

розробку ПЗ. Керуючись даними з даного розділу можна приступити до розробки ПЗ, на основі функціоналу, передбаченому у даному розділі.

3 РЕАЛІЗАЦІЯ АЛГОРИТМІЧНОГО І ПРОГРАМНОГО РІВНЯ

3.1 Розробка алгоритму

Для рослини важливо мати підтримку на всіх етапах життєдіяльності, тому система не буде закінчуватись тільки на статичному підтриманні умов. Для більш продуктивного росту потрібно встановлювати свою температуру, свою кількість добрив та рівень кислотності на кожному етапі росту. Задля цього треба створити структуру для динамічно змінюваних параметрів. У подальшому такій структурі буде надано назву “Сценарій вирощування”. Для створення моделі сценарію необхідно підготувати список параметрів, з якими працює система.

Система впливатиме на:

- температура повітря;
- світловий день;
- вологість ґрунту;
- розчин для зрошування ґрунту:
 - кількість добрив;
 - рівень PH;
 - кількість одного зрошування.

Система замірятиме:

- температура повітря;
- вологість повітря;
- рівень CO₂;
- температуру води;
- вологість ґрунту.

Для створення сценарію буде запозичено секцію “Система впливатиме на” із наведеного вище списку. Сценарій буде складатись з етапів росту, у яких всередині є дані для кожного етапу. Сценарій буде масивом одновимірних масивів, у яких і будуть зберігатись параметри для кожного з етапів.

Візуалізація сценарію зображена на рисунку 3.1.

Cycles								
ADD CYCLE								
Cycle name	Date from	Date to	Humidity threshold	Desirable PH level	Potion1	Potion2	Potion3	Manage
1	0	0	0	0	0	0	0	DELETE CYCLE
1	0	0	0	0	0	0	0	DELETE CYCLE
2	0	0	0	0	0	0	0	DELETE CYCLE
3	0	0	0	0	0	0	0	DELETE CYCLE
4	0	0	0	0	0	0	0	DELETE CYCLE
5	0	0	0	0	0	0	0	DELETE CYCLE
6	0	0	0	0	0	0	0	DELETE CYCLE

Save

Рисунок 3.1 – Візуалізація сценарію

Як можна побачити, у кожного етапу сценарію буде дата початку та дата завершення. На кожному етапі, завдяки унікальним параметрам, система буде знати, який стан системи треба прийняти. На відміну від створення системи на таймерах та вузько-направлених рішеннях, таких як окремий модуль підтримки кислотності, описане вище власне рішення може бути гнучким. Таймер – розетка (наприклад), яка може бути використана для ввімкнення освітлення, зазвичай, має ліміт 7 днів. Звичайна система підтримки кислотності взагалі навряд чи має механізм сценарію.

Для коректного функціонування системі необхідно виконати декілька циклів. Першим з таких є калібрувальний. Він відбувається перед початком запуску вирощування рослини. Калібрувальний цикл потрібен для перевірки води, чи підходить вона згідно заданих параметрів. Також під час виконання цього циклу відбувається прокачка усіх рідин, включаючи добрива, щоб уникнути утворення повітряних пробок. Даний цикл проходить згідно калібрувальному алгоритму. Цикл ініціюється при першому або повному налаштуванні пристрою. Особливістю циклу є визначення параметрів `Ph_level_in_clear_water` і `clear_water_es`. На підставі значення цих параметрів визначається якість води і можливість роботи з нею в подальшому. Значення параметра `clear_water_es` бере участь в розрахунках в основному алгоритмі

до наступного доливання води в бак з чистою водою. Алгоритм зображено у додатку В.

Далі наведено перший цикл періоду, особливістю якого є розрахунок параметрів періоду таких як: `clear_water_ec`, `water_potion_ec`, `reference`.

Крок 1. Подача чистої води в камеру для замішування згідно параметру `water_level_in_mix_camera_desirable`.

Крок 2. Вимірювання рівня ЕС в чистій воді, результативне значення записується в параметр `clear_water_ec`.

Крок 3. Подача добрив згідно параметрам `gp1`, `gp2`, `gp3`.

Крок 4. Вимірювання рівня ЕС, результативне значення записується в параметр `water_potion_ec`.

Крок 5. Розрахунок ЕС параметра `reference` по формулі: `water_potion_ec` (ЕС вода з добривами) – `clear_water_ec` (ЕС чистої води).

Крок 6. Замір рівня РН і визначення параметра `Ph_level_in_mixed_water_before_ph`.

Крок 7. Порівняння значення параметра `Ph_level_in_mixed_water_before_ph` з `ph_parametr`. Якщо рівень РН в рідині необхідно коригувати -> Крок 7.1. Якщо необхідно Крок 8.

Крок 7.1 Коригування рівня РН подачею порції об'ємом 0,1 мл необхідного розчину "+" / "-".

Крок 7.2 Запис показника використаного обсягу розчину "+" / "-" в параметр `Ph_minus_portion` або `Ph_plus_portion`.

Крок 7.3. Порівняння значення параметра `Ph_level_in_mixed_water_before_ph` з `ph_parametr`. Якщо рівень РН в рідині необхідно коригувати -> Крок 7.1. Якщо необхідно Крок 8

Крок 8. Замір рівня ЕС і запис показника в параметр `output_water_potion_ph_ec`.

Крок 9. Закачування рідини для поливу.

Розрахунки, які необхідні для функціонування системи. Для початку визначаються необхідні розрахункові значення:

– чиста вода;

- порція добрив (мл);
- вода + добрива;
- вода + добрива + pH;
- вплив порції добрива на V води;
- вплив порції Ph на V води;
- встановлена порція добрив (мл);
- встановлена V води для поливу в циклі (мл).

3.2 Конструювання програмного забезпечення

Розроблювана програмна частина містить багато класів та методів, зображених на діаграмі класів зображених у додатку Д, які будуть описані нижче у таблицях для розкриття сутності роботи програми.

В таблиці 3.1 описано клас WaterPump який описує сутність насосу.

Таблиця 3.1 – Опис класу WaterPump

Метод	Функції
pumpLitres	Прокачує необхідну кількість води. Базується на продуктивності за хв., для вмикання насосу на потрібний проміжок часу.
pumpInUntilLtres	Заповнює камеру для змішування необхідною кількістю води базуючись на показаннях датчика рівня води у камері.
pumpOutUntilLitres	Видаляє воду з камери для змішування базуючись на показаннях датчика рівня води у камері.

writeLog	Записує дані про себе у лог, що було прокачано якусь кількість води.
----------	--

В таблиці 3.2 описано клас PeristalticPump який описує сутність перистальтичної помпи.

Таблиця 3.2 – Опис класу PeristalticPump

Метод	Функції
pumpMI	Прокачує необхідну кількість добрив. Базується на продуктивності за хв., для вмикання насосу на потрібний проміжок часу.
writeLog	Записує дані про себе у лог, що було прокачано якусь кількість добрив.

В таблиці 3.3 описано клас RealTimeClock який описує сутність годинника реального часу.

Таблиця 3.3 – Опис класу RealTimeClock

Метод	Функції
getTimeString	Повертає строкове представлення часу.

getMinutes	Повертає хвилини.
getHours	Повертає години.
getDays	Повертає дні.
getMonthes	Повертає місяці.
getYears	Повертає роки.
setDateTime	Встановлює дату вказану у параметрі. Використовується для синхронізування з NTP – network time protocol.
getDifferenceWithNow	Використовується для вирахування різниці між “зараз” та заданою у параметрі датою.
nowIsBetweenTwoDates	Перевіряє чи є “зараз” у проміжку наданому у параметрах. Потрібно для виконання сценарію динамічної зміни умов росту.

В таблиці 3.4 описано клас MixCamera який описує сутність камери для замішування. Присутні наступні методи: getWaterLevel, getPhLevel, getECLevel, getState, getTemperature, adjustPH, adjustEC, normallizeWaterLevel, cleanUp. getWaterLevel зчитує кількість води у камері використовуючи датчик рівня води присутній у системі.

Таблиця 3.4 – Опис класу MixCamera

Метод	Функції
getWaterLevel	Повертає кількість води у камері використовуючи датчик рівня (CameraLevelSensor::class).
getPhLevel	Повертає рівень PH у камері використовуючи клас PHMeter.
getECLevel	Повертає рівень ЕС у камері використовуючи клас ECMeter
getState	У камери можуть бути різні стани. Щоб не сталася помилка алгоритму керуючому камерою потрібно знати стан камери. Станами можуть бути: заміряє PH, заміряє ЕЦ, заповнюється, простоє(вільна) і тд.
getTemperature	Завдяки вбудованому датчику температури води камера може виміряти її. Використовується клас WaterTemperatureSensor.
adjustPH	Зводить PH у допустимі рамки. Цикл обмежений деяким числом з конфігурацій пристрою, заміряє PH. Якщо PH менше ніж заданий, проводиться прокачка насосом класу PeristalticPump який відповідає за добавку PH плюс. Відповідно якщо PH більше заданого використовується екземпляр класу PeristalticPump закріпленний за розчином PH мінус. Цикл мікродозами намагається дійти до заданого PH і якщо за число сброб задане у конфігурації досі не вирівнено, цикл зупиняється. Це зроблено для того, щоб цикл

	не був безкінечним. В кінці кінців за N спроб алгоритм повинен привести PH в норму. Якщо цього не стається, це триггер для помилки. Система зупиниться та негайно повідомить про помилку через API.
adjustEC	Виконує основний алгоритм роботи системи. Основний алгоритм описаний вище.
normallizeWaterLevel	Яким би рівень у камері не був, камера має змогу закачувати чисту воду з резервуару чистої води, чи зпустошувати свій зміст у бак для відходів. Ця функція зробить необхідний рівень рідини був він більше чи менше.
cleanUp	Є ситуації коли камера повинна бути чистою настільки, наскільки це можливо. Даний сервісний метод дозволяє промити камеру від залишків для подальшого аналізу, наприклад, чистої води.

3.3 Архітектура програмного забезпечення

3.3.1 Стек технологій для виконання роботи

Зараз є багато способів запрограмувати мікроконтролер. Найбільш поширеною мовою для програмування вбудованих систем є мова C/C++.[11] У випадку з ESP8266, є варіант використовувати Python, але у рамках МД це не буде використано. Для зручності, буде використано середовище Arduino.

Мова програмування Arduino іменується Arduino C і являє собою мову C++ з фреймворком Wiring. Він містить деякі відмінності за частиною написання коду, який будується і компілюється з підтримкою avr-gcc. Є особливості, які полегшують написання програми – наявний комплект бібліотек, що включає в себе функції і об'єкти. При компіляції програми IDE робить короточасний файл з розширенням *.cpp. Вагомими фактором для вибору цієї платформи є підтримка плати ATmega2560

та плати ESP8266, наявність бібліотек з відкритою ліцензією для периферії, що використовується для розроблюваної системи. Відкритість платформи дозволяє використовувати її не порушуючи авторських прав а також спрощує написання коду.

3.3.2 Операційна система FreeRTOS

FreeRTOS – багатозадачна операційна система реального часу (ОСРВ) для вбудованих систем. Портована на 35 мікропроцесорних архітектур. Поширюється під ліцензією MIT з 2017 року. До 2017 року поширювалася під зміненою ліцензією GPL з виключенням, що дозволяє творцю привласнити змінений код операційної системи.

FreeRTOS створена для роботи на глобальних мікроконтролерах, з низькою швидкодією, маленьким обсягом оперативної пам'яті, відсутніми приладами управління пам'яті і реалізованих на апаратному рівні пристроїв підтримки багатозадачності та швидкого перемикавання контексту.

Диспетчер системи досить нескладний і компактний (займає, в залежності від платформи і опцій ядра, 4-9 кілобайт), втім має підтримку процесів, кооперативну багатозадачність, семафори і черги.

3.3.3 Потоки FreeRTOS

Система має багато процесів, які мають відбуватися незалежно один від одного. У даній операційній системі є можливість задати потокову функцію, яка виконуватиметься періодично з заданим інтервалом. Розрядна здатність таймеру системи сягає 15мс. Цього цілком достатньо, адже це система м'якого реального часу. Потоки також дозволяють мати як параметр пріоритети. Це означає, що потік з високим пріоритетом ніяк не може бути перерваний потоком з низьким пріоритетом а потік з високим пріоритетом може перервати потік з низьким. У системі наявні такі випадки, коли один потік може перервати інший. Наприклад існує потік, який обслуговує обмін даними із сервером. Обмін даними із сервером важлива задача, але

це така важлива як головна задача системи – підтримати задані параметри. Отже, якщо зафіксовано відхилення від заданих параметрів, система може призупинити потік передачі даних, записуючи їх до кешу, та почати приведення стану параметрів до норми. Але, як не дивно, підтримка параметрів системи, не найважливіша задача. Рослині не потрібно підтримка параметрів з точністю до секунди.

У системі є системи захисту, які, наприклад слідкують за кількістю води у камері для замішування. Цей показник дуже важливий, тому що не можна допустити її переповнення у разі несправного насосу, чи датчику рівня води у камері, або клапану, який перериває самовільне перетікання води між ємностями. Тому цей потік буде з найвищим пріоритетом, та з найшвидшою частотою виконання. Також існуює потік приймання даних від сервера. Щоб система складала враження миттєвого відгуку, потік прийому даних має 2 пріоритет, після потоку захисту пристрою.

Операційна система FreeRTOS[12] має власну функцію затримки. `vTaskDelay()` – функцію неблокуючої затримки. Це означає що при її виклику, буде блоковано лише потік, у якому вона була викликана. Це дуже зручно наприклад для вмикання реле на деякий час. Для того щоб увімкнути реле достатньо породити потік, у якому визвати послідовно функцію вмикання реле, `vTaskDelay(параметр)`, функцію вимикання реле. Інші потоки залишаться недоторканими. Після виконання потокової функції з вмикання реле, можна видалити потік, для звільнення оперативної пам'яті та інших ресурсів іншим потокам.

Ключовою характеристикою ОСРВ є рівень її послідовності у використанні кількості часу, необхідного для прийняття і виконання завдання програми; мінливість – це «джиттер». «Жорстка» операційна система реального часу (Hard RTOS) має менший джиттер, ніж «м'яка» операційна система реального часу (Soft RTOS). Пізна відповідь – це неправильна відповідь в жорсткій RTOS, в той час як пізній відповідь допустимо в м'якій RTOS. Головна мета – невисока пропускну здатність, а швидше гарантія категорії м'яких або жорстких характеристик. ОСРВ, яка зазвичай або в цілому може вкластися в термін, є ОС реального часу, але якщо вона може вкластися в термін детерміновано, то це ОС жорсткого реального часу.

RTOS має просунутий алгоритм планування. Гнучкість планувальника забезпечує більш широку оркестровку комп'ютерної системи пріоритетів процесів, але ОС реального часу частіше призначена для вузького набору додатків. Ключові фактори в ОС реального часу – мінімальна затримка переривання і мінімальна затримка перемикавання потоків. ОС реального часу цінується більше за те, наскільки швидко і передбачувано вона може реагувати, ніж за обсяг роботи, яку вона може виконати за певний період часу.

3.3.4 Відмовостійкість та системи захисту

Система підтримує задані параметри розчину для поливу і це є ідеальним варіантом її роботи. За умови, що кожен елемент системи є справним, система може змінювати РН, ЕС, кількість розчину для поливу. Буде розглянуто приклад для розуміння проблеми відмовостійкості. Датчик РН вимірює рівень кислотності, але він не у діапазоні вимірювання. З цього можна судити про вихід з ладу системи заміру рівня кислотності. Не важливо який елемент системи несправний, провід, транзистор, резистор, конденсатор, коннектор яким з'єднано датчик з платою, чи помилка програмного забезпечення. Датчик вважається несправним, що накладає наступні обмеження на роботу системи в цілому. Система, у випадку несправності такого типу, може продовжити роботу у двох варіантах. Кислотність води на момент заповнення баку чистої води, була у рамках, дозволених для поливу рослин, або без цієї перевірки, тому що в самому поганому випадку рослини поливають взагалі без контролю цього параметра. Кислотність на момент заповнення баку зберігається у енергонезалежній пам'яті – EEPROM, яка використовується для зберігання стану системи при вимиканні живлення. Отже, система все одно зробить свою задачу – зрошить рослину, але з дещо зменшеною функціональністю – без контролю кислотної. Якщо не зробити можливим полив у стані обмеженої працездатності, рослина помре не раніше, ніж у випадку, коли не поливати її зовсім. Тим паче у цьому випадку система продовжує виконувати подачу добрив.

Щодо додавання добрив до розчину для поливу. Ситуація з відмовившим датчиком кислотності не дозволяє працювати з рідинами коректорами, тому що немає рекомендованих кількостей коректорів для рослин. Є рекомендована кислотність, яку можна підвести до потрібної, спираючись на датчик. У добрив є рекомендована кількість. На кожному етапі росту, у системі керування, є поле для рекомендованих значень кожного добрива та поле бажаного ЕС. У ідеальному варіанті, коли датчик ЕС повертає свої паспортні значення, та вважається справним, параметр ЕС встановлюється у норму за рахунок спирання на показ датчика на даний момент мікродозами у пропорціях, які вказані у полях рекомендованих значень добрив. Параметр УС важливіший за кількість добрив. Якщо допустити ситуацію, коли, було добавлено рекомендовану кількість добрив, рівень ЕС не встановився у норму, це не правильно. Але все ж таки має право на існування, тому що виробники добрив рекомендують за відсутності контролю ЕС просто додавати задані кількості добрив у грамах. Звідси можна розробити стратегію роботи системау за несправного датчика електропровідності води. Так як перистальтичні насоси, у представленні програмного забезпечення розробленого у рамках даної системи, є об'єктами з методами такими як `pumpLitres()`, `pumpMlitres()`, `runMsec()`, можна зробити висновок, що є можливість прокачувати задані кількості добрив. Якщо у випадку нормальної роботи, система створює розчин спираючись на покази ЕС метра, то у даному випадку додає рекомендовані кількості кожного добрива із сценарію, що теж є допустимим та кращим, ніж зовсім не додати добрива до розчину.

Усі операції, у яких приймають участь насоси, відбуваються через МКСК схема якої показана у додатку Ж. Кожен насос може бути контрольований датчиком рівня води у МКСК. Якщо при переміщенні рідини з чи у МКСК рівень не змінюється, при умові справного датчика рівня, можна констатувати що один з насосів, чи один із елементів, що задіяний у алгоритмі прокачки рідини насосами, несправний. Таким чином, продовжувати роботу категорично заборонено. Система оперує рідинами, тому якщо рівень рідини, за умови насосу, не змінюється, можна допустити, що у найгіршому випадку, могли зіпсуватися патрубки. Це означатиме, що система може залити рідиною все навколо. Щоб цього не сталося, система повинна зупинитись, та

якщо зламане коло не насосу зливу відходів, відкачати усі рідини з усіх ємностей, та встати на паузу, чекаючи на сервісний візит. Цього достатньо щоб зупинити систему у випадку серйозної несправності насосу чи патрубків. Якщо ж несправність незначна, вона може бути непомітна такою системою захисту. Для цього, у піддоні системи, встановлений поплавков. Це рішення запозичено з миючих машин для посуду. Усе обладнання знаходиться над піддоном. Якщо у піддоні достатньо води, що підняти поплавок на замкнуті контакти – це означає протік води. Така ж система грає роль другого фактору для детектування аварії окрім системи контролю за зміною рівня води у МКСК.

Система з поплавком, зображеним на рисунку 3.2, у піддоні необхідна, тому що може бути несправний датчик рівня води у мікс камері. Датчиком рівня води є тензодатчик, який загалом використовується як ваги. Його дискретність дозволяє детектувати навіть дози добрив, які зазвичай не перевищують 1-3 мл на літр.

Система детектування та захисту МКСК від переливу, являє собою мінімум два датчика поплавка.



Рисунок 3.2 – Поплавок

Датчики поплавки, зображені на рисунку 3.3, встановлюються у нижній точці камери, та у верхній. Це потрібно для детектування верхньої та нижньої точок.

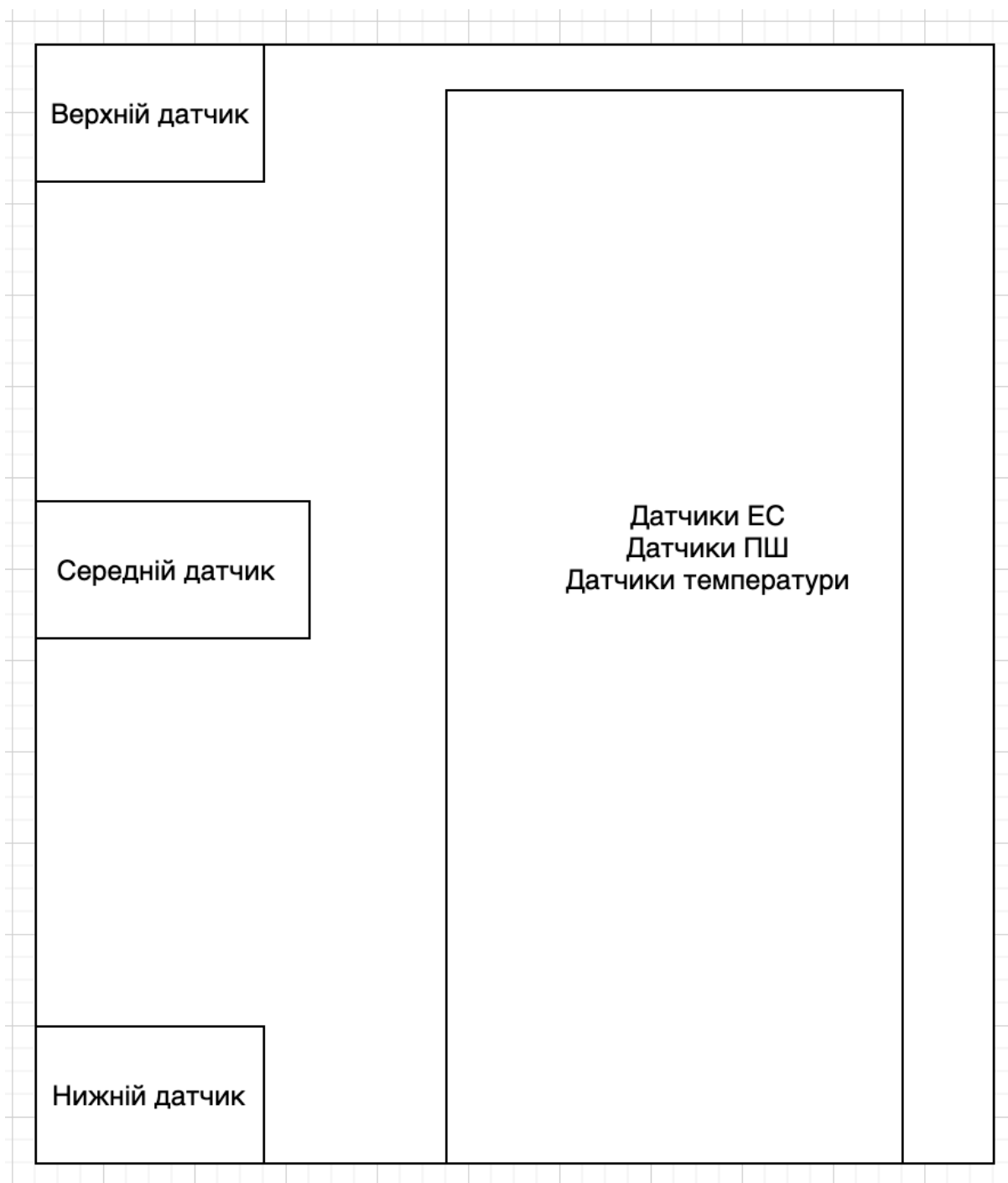


Рисунок 3.3 – Положення датчиків-поплавків

Верхній датчик потрібен для вмикання екстреного відкачування рідини у БВ. Цей датчик має двофакторний вплив на включення насоса відкачки рідини у БВ. Датчик є нормально розімкненим. Датчик одним піном підключений до входу МК, іншим до опірної напруги. У потоці захисту системи, перевірка напруги на піні спрацьовує першим пріоритетом, та запускає відкачку рідини. Але, це за умови справного мікроконтролера. МКСК це слабе місце системи через те, що вона не

герметична, тому переливання цього модулю призведе до катастрофи. Через це датчик-поплавок також підключений до входу запасного реле напругу, минуючи МК, що дозволяє за відмови мікроконтролера увімкнути відкачку рідини.

Тензодатчик потрібно іноді калібрувати. Тому, щоб це робити, потрібно додати ще один датчик між граничними.

За допомогою середнього датчика, встановленого у заздалегідь відомому місці, можна стверджувати, що у камері задана кількість рідини. У момент спрацювання датчика-поплавок можна калібрувати тензодатчик.

Для вимірювання електропровідності та кислотності суміші потрібно знати температуру розчину. У разі несправного датчика температури розміщеного у МКСК, береться значення температури вбудованого у годинник реального часу. Температура розчину та температура на годиннику реального часу може відрізнятись, але у довгостроковій перспективі температура на годиннику та температура рідин у системі буде співпадати. Тому за несправного датчика у МКСК можна брати температуру годинника.

3.3.5 Годинник реального часу та NTP

У систему присутній годинник реального часу. Він описується класом RTC. Годинник потрібен для збереження стану системи за відсутності з'єднання із мережею, або за відключення електропостачання. Годинник має власний акумулятор, завдяки чому є автономним, та збереже плин часу. Сценарій росту представляє собою часову проміжку та має бути розшифрований. Завдяки годиннику реального часу система знає, який проміжок включає в себе “зараз” і вибере його для виконання.

Протокол мережевого часу (NTP) – це мережевий протокол для синхронізації годин між комп'ютерними системами через мережі передачі даних з комутацією пакетів і змінної затримкою. Чинний з 1985 року, NTP є одним з найстаріших Інтернет-протоколів, використовуваних в даний час. NTP був розроблений Девідом Л. Міллсом з Університету Делавера.

NTP призначений для синхронізації всіх учасників комп'ютерів з точністю до декількох мілісекунд за всесвітнім координованим часом (UTC). Він використовує алгоритм перетину, модифіковану версію алгоритму Марзулло, для вибору серверів точного часу та призначений для пом'якшення наслідків ефекту змінної затримки мережі. NTP зазвичай може підтримувати час з точністю до десятків мілісекунд в загальнодоступному Інтернеті і може досягати точності вище однієї мілісекунди в локальних мережах в ідеальних умовах. Асиметричні маршрути і перевантаження мережі можуть викликати помилки тривалістю 100 мс і більше.

Протокол зазвичай описується в термінах моделі клієнт-сервер, але з такою ж легкістю може використовуватися в тимчасових відносинах, коли обидва вузла вважають один одного потенційним джерелом часу. 20 реалізацій відправляють і отримують позначки часу. з використанням протоколу призначеного для користувача (UDP) на порт номер 123. Вони також можуть використовувати трансляцію або багатоадресну розсилку, коли клієнти пасивно слухають оновлення часу після первинного каліброваного обміну туди і назад. NTP видає попередження про будь-яке майбутнє коригування секунди координації, але ніяка інформація про місцеві часових поясах або переході на літній час не передається.

Поточний протокол – версія 4 (NTPv4), який є пропонованим стандартом, як описано в RFC 5905. Він сумісний з версією 3, зазначеної в RFC 1305.

За активного підключення до мережі інтернет, система синхронізуватиме час за допомогою NTP.

3.3.6 Формат зберігання сценаріїв росту

Для зберігання сценаріїв існує флеш накопичувач. Система використовує текстовий файл як місце для зберігання сценарію. Клас SDcard описує поля і методи флеш накопичувача. Деякий з методів дозволяє читати конкретну строку у текстовому файлі. Все це дозволяє зробити висновок про формат зберігання сценарію який буде обрано. Обрано CSV.

Файл зі значеннями, розділеними комами (CSV) – це текстовий файл з роздільниками, в якому значення розділяються комами. Кожен рядок файлу являє собою запис даних. Кожен запис складається з одного або декількох полів, розділених комами. Використання коми як роздільник полів є джерелом імені для цього формату файлу. Файл CSV зазвичай зберігає табличні дані (числа і текст) у вигляді звичайного тексту, і в цьому випадку кожен рядок буде мати однакову кількість полів.

Формат файлу CSV в повному обсязі стандартизований. Основна ідея поділу полів коми ясна, але ситуація ускладнюється, коли дані поля також містять коми або вбудовані розриви рядків. Реалізації CSV можуть не обробляти такі дані поля, або вони можуть використовувати лапки, щоб укласти поле. Цитування не вирішує всього: до деяких поля можуть знадобитися вбудовані лапки, тому реалізація CSV може включати escape-символи або escape-послідовності.

Крім того, термін «CSV» також позначає кілька тісно пов'язаних форматів, розділені роздільниками, в яких використовуються інші роздільники полів, наприклад крапки з комою. До них відносяться значення, розділені табуляцією, і значення, розділені пробілами. Роздільник, наприклад табуляція, якого немає в даних поля, дозволяє спростити синтаксичний аналіз формату. Ці альтернативні файли з роздільниками-роздільниками часто з розширенням .csv, незважаючи на використання роздільник полів, відмінного від коми. Ця нечітка термінологія може викликати проблеми при обміні даними. Багато додатків, які приймають файли CSV[13], мають параметри для вибору символу роздільника і символу лапки. Крапка з комою часто використовується замість ком у багатьох європейських країнах, щоб використовувати кому як десяткового роздільника і, можливо, точку в якості десяткового символу угруповання. З цієї причини термін «значення, розділені символами» пропонується як більш широке визначення цього формату файлу.

Флеш пам'ять використовується для достатньо великих об'ємів даних. Так, кожна дія системи логується у флеш пам'ять, з метою відладки обладнання. Кожен насос має свій унікальний номер, тому у першу колонку файлу можна записувати номер насосу для ідентифікації.

3.3.7 Зберігання даних у EEPROM

EEPROM позначає електрично стираєму програмовану постійну пам'ять і являє собою тип енергонезалежної пам'яті, використовуваної в комп'ютерах, інтегрованої в мікроконтролери для смарт-карт і віддалених систем без ключа, а також інших електронних пристроїв для зберігання відносно невеликих обсягів даних, але дозволяючи стирати і перепрограмувати окремі байти.

EEPROM організовані як масиви транзисторів з плаваючим затвором. EEPROM можна програмувати і стирати всередині схеми, застосовуючи спеціальні програмні сигнали. Спочатку EEPROM були обмежені однобайтовими операціями, що робило їх повільніше, але сучасні EEPROM допускають багатобайтові операції зі сторінками. EEPROM має обмежений термін служби для стирання і перепрограмування, тепер він сягає мільйона операцій в сучасних EEPROM. У EEPROM, який часто перепрограмується, термін служби EEPROM є важливим аспектом проектування.

Флеш-пам'ять – це тип EEPROM, розроблений для забезпечення високої швидкості і високої щільності за рахунок великих блоків стирання (зазвичай 512 байт або більше) і обмеженої кількості циклів запису (часто 10 000). Між ними немає чіткої межі, але термін «EEPROM» зазвичай використовується для опису незалежної пам'яті з невеликими блоками стирання (розміром всього один байт) і тривалим терміном служби (зазвичай 1000000 циклів). Багато мікроконтролерів включають в себе як флеш-пам'ять для прошивки, так і невелику EEPROM для параметрів і історії.

Станом на 2020 рік флеш-пам'ять коштує набагато менше, ніж байтово-програмована EEPROM, і є домінуючим типом пам'яті там, де системі потрібно значний обсяг незалежної твердотільної пам'яті. EEPROM, однак, все ще використовуються в додатках, яким потрібно лише невеликий обсяг пам'яті, наприклад, при послідовному виявленні присутності.

Для подовження ресурсу роботи флеш-пам'яті, клас SDcard має функцію кешування значень у EEPROM. Для цього існує допоміжна функція яка може розміщувати такі типи даних у пам'яті як integer і double. Ціле число можна умістити

у два байти, в ту чергу як double займає 4 байти. Це потрібно пам'ятати при виділенні пам'яті під ті чи інші значення.

В один момент часу система може мати стан тільки одного елементу сценарію. У елементі сценарію є до 10 параметрів. Отже, для їх кешування потрібно потенційно до 40 байтів. Також, є деякі параметри які система має заміряти та запам'ятовувати. Це кислотність чистої води та електропровідність. Ця інформація потрібна для подальшої роботи головного алгоритму приготування суміші.

Для зручності аллокування параметрів у пам'яті було створено класи обгортки для даних які зберігаються у EEPROM.

```
EEPROMDoubleoption usedPotion1(72);
```

```
EEPROMDoubleoption usedPotion2(76);
```

```
EEPROMDoubleoption usedPotion3(80);
```

```
EEPROMDoubleoption usedPotion4(84);
```

```
EEPROMDoubleoption usedPotion5(88);
```

```
EEPROMDoubleoption presensePotion1(204);
```

```
EEPROMDoubleoption presensePotion2(208);
```

```
EEPROMDoubleoption presensePotion3(212);
```

```
EEPROMDoubleoption presensePotion4(216);
```

```
EEPROMDoubleoption presensePotion5(220);
```

```
EEPROMDoubleoption usedTrashbox(106);
```

```
EEPROMDoubleoption usedClearWater(60);
```

```
EEPROMDoubleoption clear_water_ec(102);
```

```
EEPROMDoubleoption clear_water_ph(110);
```

```
EEPROMDoubleoption trashboxVolume(224);
```

```
//notifications
```

```
//white
```

```
StateEvent wateringHappenedEvent(1, 185, 2, 3, "wateringHappenedEvent");
```

```
StateEvent growingIsFinishedEvent(1, 185, 4, 5, "growingIsFinishedEvent");
```

```
StateEvent growingIsStartedEvent(1, 185, 6, 7, "growingIsStartedEvent");
```

```
StateEvent growingPeriodIsDoneEvent(1, 186, 0, 1, "growingPeriodIsDoneEvent");
```

```
//yellow
```

```
StateEvent lessWaterEvent(2, 184, 2, 3, "lessWaterEvent");
```

```
StateEvent newWaterEvent(2, 184, 4, 5, "newWaterEvent", "checkWaterRequest");
```

```
StateEvent trashIsAlmostFullEvent(2, 186, 2, 3, "trashIsAlmostFullEvent");
```

```
StateEvent potion1IsAlmostEmptyEvent(2, 186, 4, 5,
"potion1IsAlmostEmptyEvent");
```

```
StateEvent potion2IsAlmostEmptyEvent(2, 186, 6, 7,
"potion2IsAlmostEmptyEvent");
```

```
StateEvent potion3IsAlmostEmptyEvent(2, 187, 0, 1,
"potion3IsAlmostEmptyEvent");
```

```
StateEvent potion4IsAlmostEmptyEvent(2, 187, 2, 3,
"potion4IsAlmostEmptyEvent");
```

```
StateEvent potion5IsAlmostEmptyEvent(2, 187, 4, 5,
"potion5IsAlmostEmptyEvent");
```

```
StateEvent ECx2WarningEvent(2, 189, 0, 1, "ECx2WarningEvent");
```

```
//StateEvent trashIsAlmostFullEvent(2, 186, 2, 3, "trashIsAlmostFullEvent");
```

```
//red
```

```
StateEvent noWaterEvent(3, 184, 0, 1, "noWaterEvent", "checkWaterRequest");
```

```
//red error
```

```
StateEvent trashIsFullEvent(3, 184, 6, 7, "trashIsFullEvent", "trashIsEmpty");
```

```
StateEvent drainageIsFullEvent(3, 185, 0, 1, "drainageIsFullEvent");
```



```

    StateEvent potion1IsEmptyEvent(3, 187, 6, 7, "potion1IsEmptyEvent",
"managePotions");

    StateEvent potion2IsEmptyEvent(3, 188, 0, 1, "potion2IsEmptyEvent",
"managePotions");

    StateEvent potion3IsEmptyEvent(3, 188, 2, 3, "potion3IsEmptyEvent",
"managePotions");

    //ph minus state event

    StateEvent potion4IsEmptyEvent(3, 188, 4, 5, "potion4IsEmptyEvent",
"managePotions");

    //ph plus state event

    StateEvent potion5IsEmptyEvent(3, 188, 6, 7, "potion5IsEmptyEvent",
"managePotions");


    //black

    StateEvent clearPumpIsBroken(4, 189, 2, 3, "clearPumpIsBroken");

    //black error

    StateEvent trashPumpIsBroken(4, 189, 4, 5, "trashPumpIsBroken");

    //black error

    StateEvent phCorrectionIsBroken(4, 189, 6, 7, "phCorrectionIsBroken");
    StateEvent ecCorrectionIsBroken(4, 190, 0, 1, "ecCorrectionIsBroken");
    StateEvent outPumpIsBroken(4, 190, 2, 3, "outPumpIsBroken");

```

Вище у прикладі наведені декларації об'єктів для кожного параметру, що зберігається у пам'яті. У класу EEPROMDoubleoption у параметрі вказується адреса зберігання даних. Врахуйте та зверніть увагу що крок дорівнює 4. StateEvent використовує пам'ять дещо інакше. У строці StateEvent inPumpIsBroken(4, 190, 4, 5, "outPumpIsBroken") 190 – номер байту, 4 – п'ятий біт байту 190, 5 – шостий біт байту 190. У строці StateEvent outPumpIsBroken(4, 190, 2, 3, "outPumpIsBroken") 190 – номер байту, 2 – третій біт байту 190, 3 – четвертий біт байту 190. Таким чином цей клас використовує лиш два біта з байту, що дозволяє в подальшому написати наступну

конструкцію: `StateEvent ecCorrectionIsBroken(4, 190, 0, 1, "ecCorrectionIsBroken");`, тому що перший і другий біти у цьому байті ще не зайняті і не використовуються.

3.4 Метод підключення до мережі

Плата ESP8266 має змогу підключення до Wi-Fi 2.4GHz. Плата підтримує режими роботи такі як: клієнт, точка доступу, одночасно клієнт і точка доступу. Для відповідності до вимог системи управління, та можливості працювати через мережу Інтернет та протокол WebSockets, плата має бути підключена до мережі Wi-Fi з наявним доступом до глобальної мережі – Інтернет. Щоб цього досягти, потрібно розробити інтерфейс, частково зображений на рисунку 3.5 та повністю у додатку Г, для первинного налаштування плати, та підключення останньої до мережі Інтернет.

Винайдено рішення транслювати власну точку доступу. Це працює дякуючи тому, що плата має змогу працювати у режимі AP-STA, що означає Access Point – Station. В той час, як у одній бездротовій мережі з назвою “Device 1 AP” працює сервер, який може обробляти запити по протоколу HTTP, плата може сканувати інші, крім своєї, наявні мережі поблизу.

Для власної точки доступу написано веб-сервер, який може обробляти деякі запити. Та розроблено інтерфейс.

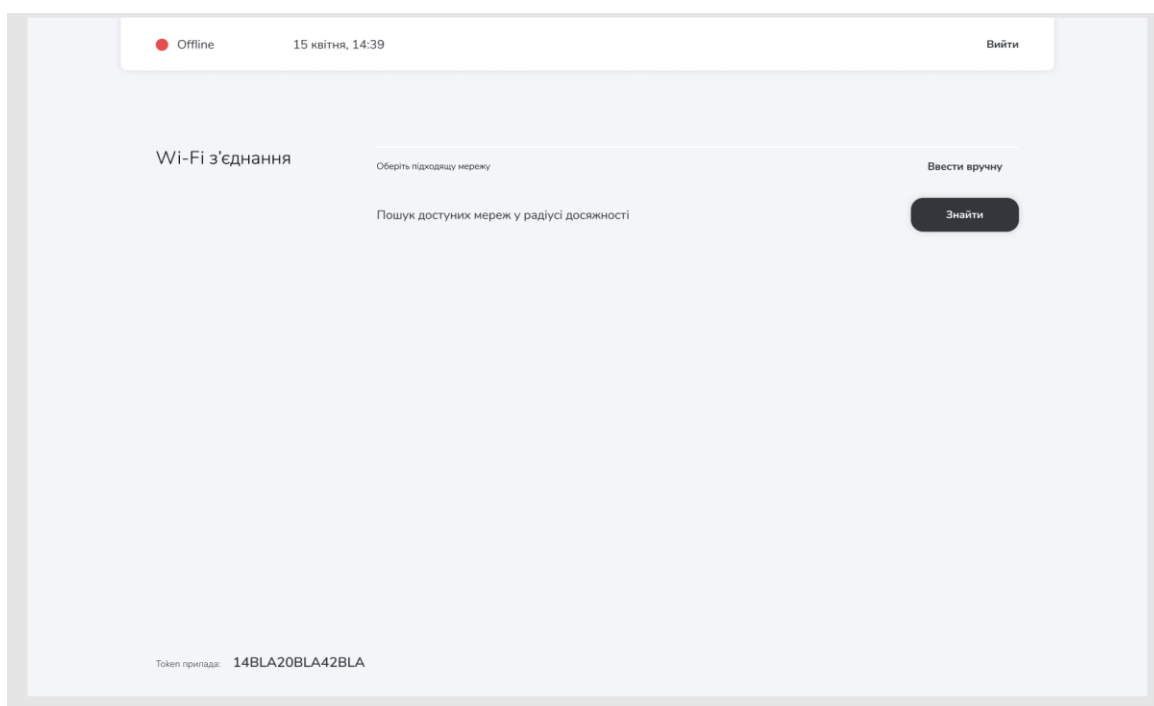


Рисунок 3.4 – Інтерфейс

На даному етапі виконується пошук можливих мереж поблизу приладу. Інтерфейс інтуїтивно зрозумілий отже не потрібно мати спеціальних навичок для підключення приладу до мережі Інтернет.

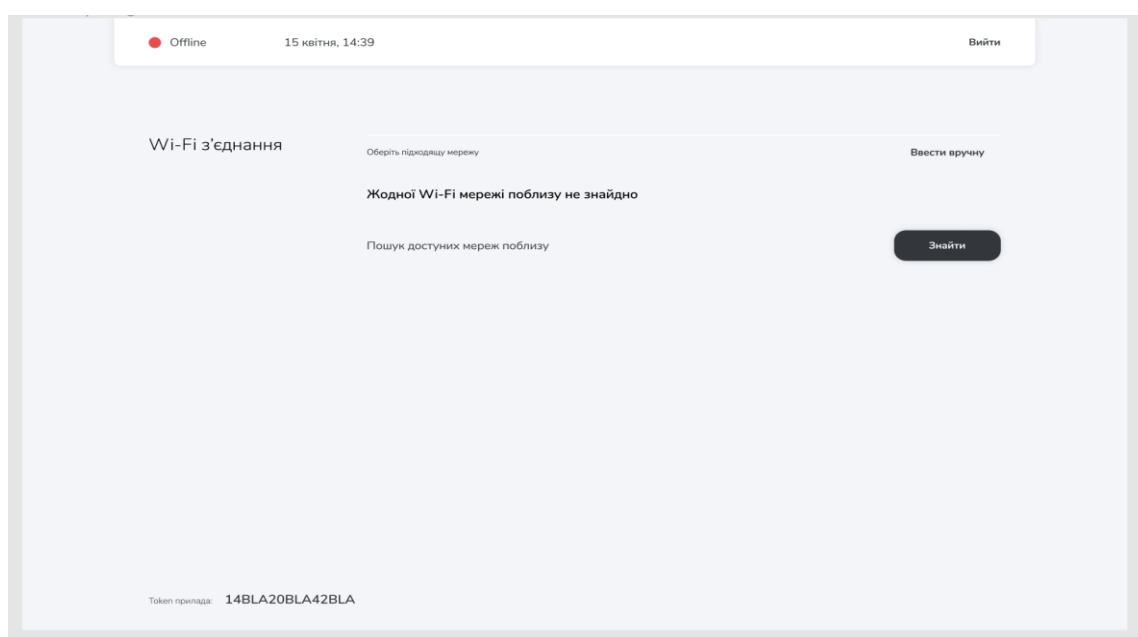


Рисунок 3.5 – Мережа не знайдена

Якщо жодної мережі не знайдено, у такому випадку отримано сповіщення про необхідність почати пошук заново, зображеного на рисунку 3.5.

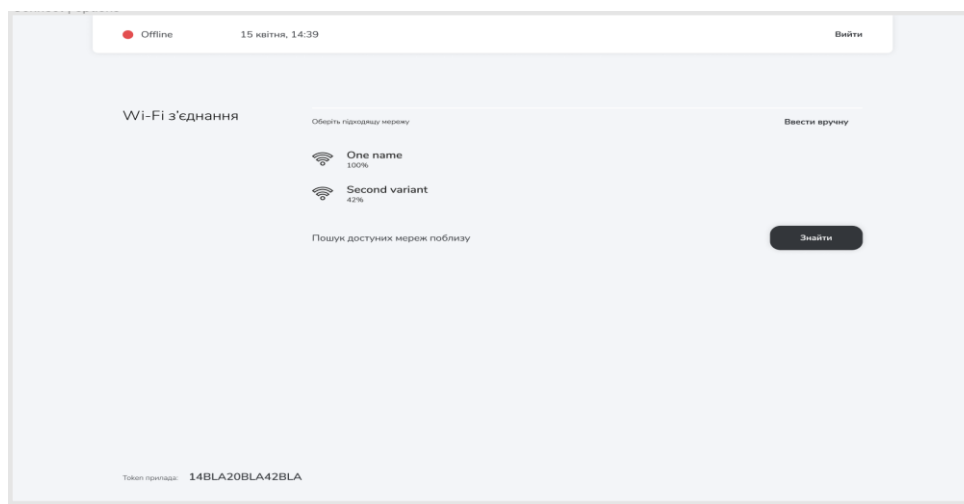


Рисунок 3.6 – Мережа знайдена

Коли врешті решт знайдено мережі, зображені на рисунку 3.6, можна підключитися до будь-якої з них. Після чого буде запитано пароль до мережі, та змінено статус на “Connecting”. Індикатор у цей час яскравіє оранжевим кольором. Щоб підключитися до мережі ще раз, достатньо ще натиснути на іконку мережі, та повторити всі дії. За натискання на строку де зазначено SSID мережі, відкриється модальне вікно, у якому можна зазначити парольну фразу. У списку, крім назви мереж існує індикатор сили сигналу. Індикатор працює наступним чином. Кожна поділлка має свій проміжок сили сигналу вимірюваного у децибел-міліватах. dBm – скорчення яке використовується для опису даної величини.

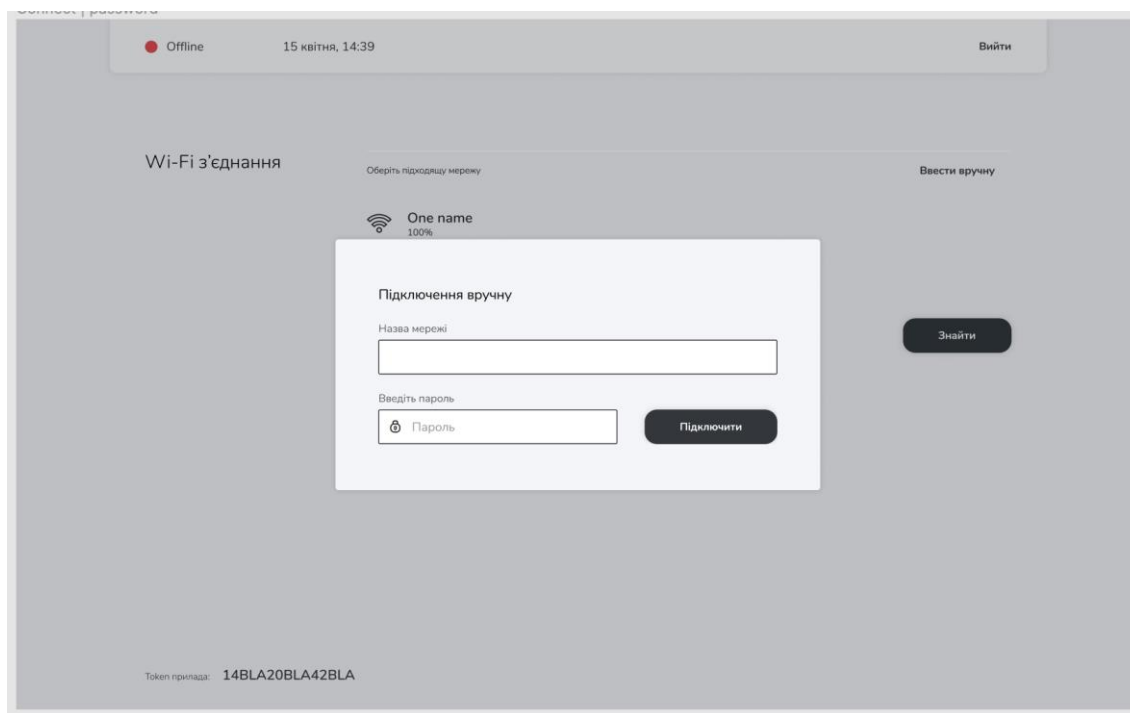


Рисунок 3.7 – Підключення вручну

У даному інтерфейсі підключення прилада до мережі, який використовує вбудовану мережу стандарту IEEE 802.11N, присутні поділки на індикаторах сили сигналу. Вони поділяються на зони такі як[14]:

- -30 дБм. Це максимальна потужність сигналу, яку ви могли досягти, і це, мабуть, означає, що ви знаходитесь на відстані декількох футів від маршрутизатора Wi-Fi;
- -50 дБм. Все, що перебуває від -30 дБм до -50 дБм, є чудовим і часто приблизно настільки високе, наскільки ви можете отримати взагалі;
- -67 дБм. Це найнижче значення, яке забезпечить прийнятні результати для більшості онлайн-послуг;
- -80 дБм. Це найнижче можливе значення, яке встановить зв'язок між маршрутизатором та клієнтом, але воно занадто слабе, щоб бути корисним для будь-якої онлайн-служби.

Процес підключення може тривати достатньо довго тому передбачено “прогресс-бар”. Зазвичай процес підключення не тривалий процес, але як і з силою сигналу, для наглядності було розроблено елемент анімації.

Індикатор статусу приладу має три вигляди: червоний, оранжевий, зелений. Червоний колір означає що пристрій не підключений до мережі і не веде діяльності спрямованої на отримання статусу “Підключено” до всесвітньої мережі Інтернет.

Описана вище система керування на первинному етапі має задачу надання змоги користувачеві підключитися до іншої мережі за допомогою вбудованої. Для цього дизайн що був представлений у рисунках даного пункту, треба перевести у зрозумілий веб браузеру формат.

Плата має у собі всього 8Мб флеш пам’яті, та не має ніякої звичайної операційної системи на рахунок Linux. Все що може робити WEB-сервер, який являє собою бібліотеку “ESP8266WebServer”, це обробляти запит за певним маршрутом на адресі.

Сервер працює за адресою 192.168.42.1. Це означає що за переходу за цією адресою буде визвано маршрут “/”.

Наступним кодом можна конфігурувати ESP8266 що запустити веб сервер:

```
Serial.setRxBufferSize(1024);
Serial.begin(115200);
WiFi.mode(WIFI_AP_STA);
wifiManager.setTimeout(1000);
wifiManager.setupConfigPortal("Corona", "Beerruss");
Serial.println("ESP Started");
client.onEvent(onEventsCallback);
client.onMessage(onMessageCallback);
```

Для зручності переходу до вбудованого інтерфейсу підключення до існуючої мережі було використано технологію Captive Portal.

Captive Portal – це веб-сторінка, яка відображається для недавно підключених користувачів Wi-Fi або проводової мережі, перш ніж їм буде надано більш широкий доступ до мережевих ресурсів.[15] Адаптивні портали зазвичай використовуються для подання угоди про використання сторінки або сторінки входу, яка може вимагати аутентифікації, оплати, використання ліцензійної угоди з кінцевим використанням,

прийнятного використання, опитування або інших діючих облікових даних, які користувач погоджує. Captive Portal використовується для широкого мобільних і стаціонарних широкосмугових послуг, включаючи кабельні та комерційні Wi-Fi і домашні точки доступу. Адаптивний портал також можна використовувати для забезпечення доступу до корпоративних або житловим провідним мережам, таким як житлові будинки, готельні номери і бізнес-центри.

Captive Portal представляється клієнту і зберігається або на шлюзі, або на веб-сервері, на якому розміщена веб-сторінка. У набір функцій шлюзу веб-сайти або порти TCP можуть бути внесені в білий список, щоб користувачеві не доводилося використовувати залежності з адаптивним порталом для їх використання. MAC-адресу підключених клієнтів також можна використовувати для обходу процесу входу в систему для зазначених пристроїв.

Існує кілька способів реалізації адаптивного порталу:

- перенаправлення HTTP;
- перенаправлення ICMP;
- перенаправлення по DNS.

Поширеним методом є перенаправлення всього трафіку WWW на веб-сервер, який повертає HTTP-перенаправлення на портал авторизації. Коли сучасний пристрій з підключенням до Інтернету вперше підключається до мережі, воно відправляє HTTP-запит на URL-адресу виявлення, попередньо визначений його постачальником, і очікує код стану HTTP 200 або 204. Якщо пристрій отримує статус HTTP 200 код, передбачається, що у нього є необмежений доступ в Інтернет. Підказки адаптивного порталу відображаються, коли ви можете маніпулювати цим першим повідомленням HTTP, щоб повернути код стану HTTP 302 (перенаправлення) на обраний вами захоплюючий портал.

Клієнтський трафік також можна перенаправити за допомогою перенаправлення ICMP на рівні 3.

Коли клієнт запитує ресурс з WWW, браузер запитує DNS. На адаптивному порталі брандмауер гарантує, що тільки DNS-сервери, надані мережевим DHCP, можуть використовуватися нерозпізнаними клієнтами (або, як альтернатива, він буде

перенаправляти всі DNS-запити нерозпізнаних клієнтів на цей DNS-сервер). Цей DNS-сервер поверне IP-адресу сторінки адаптивного порталу в результаті всіх запитів DNS. Щоб виконати перенаправлення за допомогою DNS, треба використати перехоплення DNS для виконання дії, аналогічної атаці «людина посередині». Щоб обмежити вплив зараження DNS, зазвичай використовується значення TTL, рівне 0.

У разі невдалого перехоплення першого запиту пристрою система залишається працездатною. Для того щоб перейти до порталу керування бездротовими мережами, треба використати будь-який браузер на пристрої, з якого відбувається спроба перейти до порталу вбудованого порталу керування відомими бездротовими мережами. Щоб потрапити до порталу, треба ввести адресу 192.168.43.1 на якій і розміщено веб сервер, що дозволяє керувати обліковими записами мереж Інтернет стандарту IEEE802.11n.

Щоб зрозуміти складність розміщування такого порталу на платі ESP8266 розглянуто наступний приклад:

```
void WiFiManager::getStatus() {
    DEBUG_WM(F("Info"));

    // String page = FPSTR(HTTP_HEADER);
    String page = "";
    String connected_state = "";
    String websocket_connected = "";
    String connected_ssid = "";
    String connected_attempt_color = "";
    String connected_attempt_ssid = "";

    if(_connectAttempt == 2 && WiFi.status() == WL_CONNECTED){
        connected_attempt_color = "2";//green
        connected_attempt_ssid = _ssid;
    }
```



```

if(_connectAttempt == 10){
    connected_attempt_color = "0";//red
    connected_attempt_ssid = _ssid;
}

```

```

if(_connectAttempt > 0 && _connectAttempt < 10 && _connectAttempt != 2 ||
(_connectAttempt == 2 && WiFi.status() != WL_CONNECTED)){
    connected_attempt_color = "1";//#cea308
    connected_attempt_ssid = _ssid;
}

```

```

if(WiFi.status() != WL_CONNECTED){
    if(_connectAttempt > 0 && _connectAttempt < 10){
        _connectAttempt += 1;
    }
}

```

```

if(connected == true){
    connected_state = "connected";

    if (WiFi.SSID() != "") {
        connected_attempt_ssid = WiFi.SSID();
    }
    if (_ssid != "") {
        connected_attempt_ssid = WiFi.SSID();
    }
}

```

```

if(connected_websocket){//authorized however
    websocket_connected = "1";
}

```

```

}

if(connected == false){
    connected_state = "disconnected";
    websocket_connected = "0";
    connected_ssid = "не підключена";
}

page += "{ \"e\": \"sd\", \"wc\": \"\" + websocket_connected + "\", \"ssid\": \"\" +
connected_attempt_ssid + "\", \"ca\": \"\" + _connectAttempt + "\", \"cac\": \"\" +
connected_attempt_color + \"\"}";

page += String(F("<h3 style='display: inline'>Сепер: </h3>"));
page += String("<div id='power_state' style='background-color:' + connected_color
+ ';'></div>");

page += String("<h3>Wi-Fi сеть: " + connected_ssid + "</h3>");
page += String("<div id='attempt' style='visibility:hidden;'>" +
String(_connectAttempt) + "</div>");

page += String("<div id='connected_attempt' style='border-radius:5px;background-
color:' + connected_attempt_color + ';'color: white'>" + connected_attempt + "</div>");

server->sendHeader("Access-Control-Allow-Origin", "*");
server->sendHeader("Content-Length", String(page.length()));
server->send(200, "text/html", page);

DEBUG_WM(F("Sent old info page"));

```

Наведений код використовується для виводу іконки статусу, зображену на рисунку 3.8.

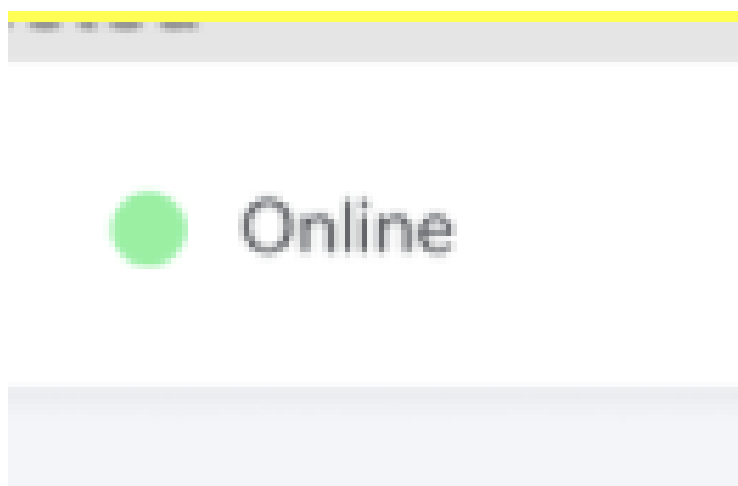


Рисунок 3.8 – Іконка статусу

Все ускладнюється тим, що звичної файлової системи немає, тому потрібно збирати сторінку у кодї на моментї написання програми і заливати разом з основною програмою як змінні. Будь-які технології, такі як Flask, .NET MVC, NodeJS, Laravel недоступні. Фреймворки не будуть працювати на даній платі, тому що плата Wi-Fi ESP8266 не є хоча б мікрокомп'ютером архітектури ARM. Вставляти фрагменти коду HTML, CSS, JS у код програми на C++ – єдиний, але робочий спосіб розмістити інтерактивну веб-сторінку для керування внутрішніми процесами плати розширення функціоналу – Wi-Fi.

3.5 Висновки по розділу

Створено алгоритм, за яким система має змінювати параметри суміші. На основі структурної, функціональної схеми, алгоритму та технічного завдання спроектовано програмне забезпечення для МК.

Розроблено механізм зберігання сценаріїв у МФК. Створено «захоплюючий портал» для первинного підключення пристрою до мережі Інтернет. Програмне забезпечення готове до завантаження у МК.

4 ТЕХНІЧНА РЕАЛІЗАЦІЯ, ТЕСТУВАННЯ АЛГОРИТМУ

4.1 Визначення переліку апаратного забезпечення

Реальне зображення датчику кислотності присутнє на рисунку 4.1.



Рисунок 4.1 – рН датчик[16]

Для реалізації вище описаної системи потрібні: мікроконтролер, силові модулі реле, насоси, датчики. Обрано наступний перелік технічного обладнання:

- мікроконтролер Atmel Mega 2560;
- шилд для 3 пінового підключення;
- ESP8266;
- антена WIFI 2.4GHz +3dBi з інтерфейсним кабелем;
- РН датчик;
- сенсор температури води;
- сенсор вологості ґрунту;

- сенсор рівня поплавков;
- датчик диму та газів MQ-2;
- датчик температури повітря;
- модуль micro SD/TF карт;
- годинник RTC DS3231;
- акумулятор LIR2032;
- 4-канальний модуль реле 5В;
- помпа перистальтична – 5 од.;
- мембранний насос (12 В, 2 л/хв) – 2 од.;
- блок живлення AC-DC понижуючий з 220В на 12В 5А (60Вт).

4.2 Компонування елементів

Для первинного запуску та наладки елементи згруповано на електромонтажному стенді. Елементи керуючого органу мають 2.54 мм JST роз'єми на стороні логіки. Для комутації «землі» з усіма елементами, використовується макетна плата MB-102 830. За замовчанням МК має 4 виводи землі, чого не вистачить для комутації більше, ніж 4 незалежних елементів, кожному з яких потрібно підключення до «землі». Для поширення напруги живлення, також використовується макетна плата.

Датчик температури рідини, з найменуванням DS18B20, з заводу не готовий виконувати свою функцію. Для функціонування потрібно встановити підтягуючий резистор [17] номіналом 4.7 кОм у розрив Data та напруги живлення як показано на рисунку 4.2. Також можливе становлення резистора номіналом 5 кОм, що було експериментально протестовано у ході виконання завдання. Усі останні елементи, такі як датчики вологості, температури повітря, кислотності розчину, електропровідності рідини, датчик вагів, датчик поплавков, датчик потоку рідини, не потребують доробки та можуть бути напряму підключені до МК. Логічні рівні у датчиків та МК співпадають. Струм який споживають датчики не перевищує максимально допустимий для МК.

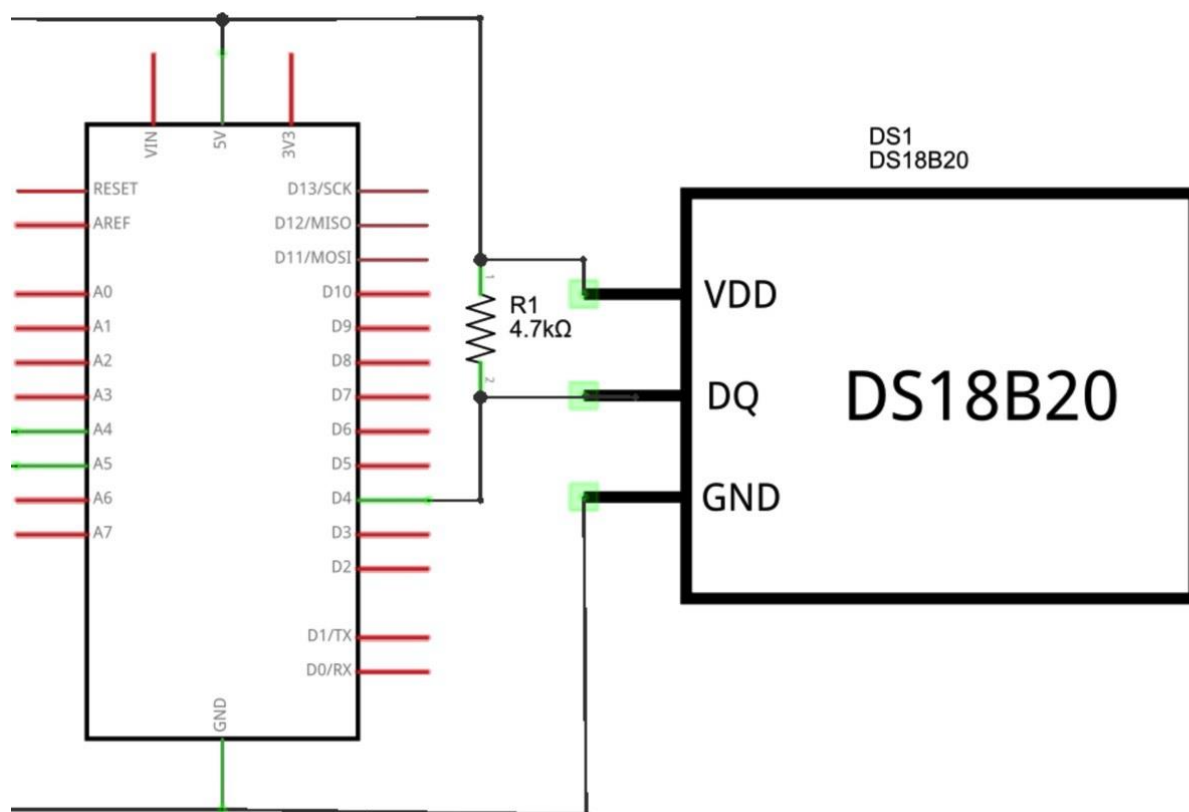


Рисунок 4.2 – Варіант підключення DS18B20

4.3 Прошивка МК

Плата налагодження Mega2560 має на борту перетворювач CH430G. Це означає, що можна підключити плату налагодження безпосередньо до ПК через USB-порт. Плата виступає у ролі імітатора фізичного послідовного порту комп'ютера, що, зазвичай, зараз зустрічається у комп'ютерах рідко. Далі USART використовується для спілкування з платою налагодження, прошивки, виводу налагоджувальних даних у консоль.

Для обміну інформацією USART використовує дві лінії: RxD і TxD. Одна лінія використовується для приймання інформації, інша – для передавання.[19] Є декілька варіантів для вибору швидкості передачі даних: 2400, 4800, 9600, 14400, 19200, 28800, 115200 бодів. Для МК та МЗ, які використовуються у МД, швидкість вибрано 115200 бод, як максимально можливу для стабільної роботи.

Для написання коду було використано Atmel Studio. Для запуску програми необхідно задовольнити наступні вимоги до ПК[18]:

- Windows 10;
- 64-bit (x64) архітектуру;
- комп'ютер з 1.6 GHz або швидшим процесором;
- оперативна пам'ять у обсязі:
 - 1 GB RAM для x86;
 - 2 GB RAM для x64;
 - додатково 512 MB RAM якщо запуск у Virtual Machine.
- 6 GB доступного місця на диску.

Для прошивки МК використовується утиліта Avrdude та STK500[20]. Її можна запустити з Atmel Studio. Наприклад, щоб загрузити скомпільований код безпосередньо за допомогою Avrdude, треба виконати наступну команду: «avrdude -p m2560 -c stk500v2 -P /path/to/serial -b 115200 -F -U flash:w:/path/to/image.hex»

4.4 Перевірка прошивки

Після того як прошивка була завантажена до мікроконтролера, можна перевіряти роботу програмного забезпечення. Первинний огляд вбудованого світлодіода плати може дати знак про те, що ПЗ було запущено та виконується.

Вбудований світлодіод, що за замовченням підключений до GPIO13 плати Mega2560, запрограмований блимати з інтервалом 1 с. Після упевнення, що відтворення програми відбувається, можна тестувати алгоритм. Тестування алгоритму проходить у два етапи: нормальний плин росту рослини, виключні ситуації. У випадку виключних ситуацій, система має відреагувати на зміни та адаптуватися до середовища, яке встановилося. Датчики температури, пологості ґрунту та температури попередньо перевірені.

4.5 Тестування алгоритму

Увімкнена система має зрошувати рослину. Для тестування у лабораторних умовах, можна прийняти, що зволоження та висихання ґрунту буде імітувати датчик вологості занурений або витягнутий зі склянки з водою. Горщиком з рослиною буде слугувати лабораторна мірна склянка об'ємом 1 л. Вона необхідна для тестування об'єму розчину, який готує система. Щоб відслідковувати зміну кислотності, використано ручні датчики кислотності та електропровідності рідини. Дані замірів занесено у таблиці наведені у рисунках.

Головними параметрами серед даних, зібраних під час тестування алгоритму, є наступні :

- n – номер заміру;
- temperature – температура суміші;
- clear_water_ec – електропровідність чистої води;
- water_potion_ec – електропровідність суміші після додавання добрив;
- reference – вплив добрив на порцію води;
- target_ec – бажана електропровідність;
- difference_ec – різниця дренажу та бажаної;
- drainage_ec – електропровідність дренажу;
- water_level_in_mix_camera_before_mixing – рівень рідини до змішування;
- water_level_in_mix_camera_desirable – рівень рідини бажаний;
- water_level_in_mix_camera_additional – рівень рідини додатковий;
- ph_before_mixing – кислотність до змішування;
- p1 – коректор 1 (мл);
- p2 – коректор 2 (мл);
- ph_level_in_clear_water – кислотність чистої води;
- output_water_lvl.- об'єм рідини для зрошування;
- rp1- передбачаємий коректор 1 (мл);
- rp2 – передбачаємий коректор 2 (мл);
- ph_plus_potion – коректор кислотності вгору (мл);
- ph_minus_potion – коректор кислотності вниз (мл);
- output_ph_lvl – кінцевий замір кислотності;

Головною метою системи є на кінцевому етапі роботи алгоритму зрівняти `output_water_potion_ph_ec` з `target_ec`, `ph_parameter` з `output_ph_lvl` та `water_level_in_mix_camera_desirable` з `output_water_lvl`. Якщо ці параметри співпадають, можна стверджувати що система працює.

Для `id 137` з таблиці на рисунку 4.2 та рисунку 4.1 параметр `output_water_lvl` дорівнює 502 мл, в той час як `water_level_in_mix_camera_desirable` дорівнює 0.5 л = 500 мл. Похибка створення необхідної кількості розчину дорівнює $502/500 * 100\% = 0.4\%$. Для виявлення більш точної похибки системи у подальшій перспективі потрібно зробити дещо більше тестів, але цих тестів було достатньо щоб перейти до налаштування інших підсистем системи.

Наприклад, для таблиць з рисунків пунктів 2 налаштований розбіг для корекції РН рівний 0.5 одиниць. Це означає, що за наявного параметра `ph_parameter`, який дорівнює 7, вихідні значення розчину не мають перевищені рамки 6.5 на нижній границі, та 7.5 на верхній границі. Якщо взяти строку з `id`, який дорівнює 72, то можна побачити, що параметр `ph_before_mixing` був спровокований, щоб бути 10,48. Це було зроблено вручну, за допомогою внесення коректора кислотності з підвищуючими характеристиками. Після чого реакція системи була наступна: параметр `output_ph_lvl` встановився на відмітці 6,947022, що є нормою[21], і входить до границь 6.5 – 7.5. Таким чином можна стверджувати, що на даному етапі підсистема підведення кислотності у границі, близькі до потрібної, виконана успішно. Інші тести можуть показати схожу тенденцію, у кожному тесті, але загалом система видає правильно параметризовану суміш для поливу.

4.6 Висновки до розділу

Після маніпуляцій, описаних у розділі, проведених із елементами системи отримано робочу систему. Завдяки цьому, перейдено до наступного кроку – тестування системи. Для тестування системи налагоджено лабораторне середовище з необхідним устаткуванням.

Завдяки наявному обладнанню та проведенням тестам отримано результати роботи системи, що задовольняють потреби. Система підтримує параметри кислотності у границях «задана – 0.5 од» < задана < «задана + 0.5 од», що є допустимим для проходження росту рослини, подає потрібні кількості добрив та кількість чистої води.

5 РОЗРОБЛЕННЯ СТАРТАП-ПРОЄКТУ

5.1 Опис ідеї проєкту

Описана у магістерській роботі системи може мати декілька реалізацій. Для створення ідеї стартап-проєкту, описаної у таблиці 5.1, у ролі кінцевого споживача обрано рядового користувача. Через сучасний підхід до розробки з метою покращення користувацького досвіду, користувачі, які бажають автоматизувати свої теплиці, але не мають спеціальним технічних навичок і будуть потенційними користувачами системи.

Таблиця 5.1 – Опис ідеї стартап-проєкту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка системи для автоматизації процесу зрошування рослини.	1. Використання добрив при зрошенні.	Використання добрив правильної позитивно впливає на ріст та розвиток рослини, а автоматизація такого процесу поліпшує результати та економить час.
	2. Зменшення помилок у виробництві.	Зменшення вагомості людського фактору підвищує результативність та майже позбавляється помилок у процесі зрошення.
	3. Діагностика показників рослини.	Користувач має змогу переглядати результати та показники, що дозволяє оперативно вносити зміни у процес. Наявність датчиків дозволяє у реальному часі контролювати показники зовнішньої середовища рослини.

	4. Групування схожих рослин.	Схожі рослини можна згрупувати і поливати в одній або декількох зонах, що економить час при зрошуванні рослини.
	5. Отримання більш однорідного урожаю.	Рівномірні, послідовні цикли поливу сприяють отриманню більш однорідного урожаю з кращими кількісними результатами.
	6. Задання сценаріїв вирощування, унікальних для рослини.	Задання унікальних сценаріїв сприяє більш продуктивному росту рослини.

Таблиця 5.2 – Характеристики ідеї проекту

№ п/ п	Техніко- економічні характеристики ідеї	Товари/концепції конкурентів				W (слабка сторона)	N (нейтральн а сторона)	S (сильна сторона)
		Мій проект	Grob o	Lea f	Cloud Ponic s			
1.	Зручність користування	10	8	8	7			+
2.	Відмовостійкіс ть системи	8	8	8	8		+	

3.	Тонке налаштування	9	6	9	5			+
4.	Відповідність ціни до якості	10	9	8	6			+
5.	Можливості розширення системи	5	1	6	2	+		
6.	Простота освоєння системи	10	9	6	8			+
7.	Дизайн фізичного апарату згідно новітнім тенденціям	8	0	9	8	+		

Провівши аналіз системи, отримано сильні та слабкі сторони системи, які перелічено у таблиці 5.2, що дозволить обрати маркетингову стратегію та вказувати на сильні сторони.

5.2 Технологічний аудит ідеї проекту

Таблиця 5.3 – Технологічна здійсненність ідей проекту

№ п/п	Ідея проекту	Технологія реалізації	Наявність технології	Доступність технології
1.	Програмування мікроконтролеру	Мова C/C++	Наявна	Доступна
		Мова Python	Наявна	Доступна

		Середовище Arduino Arduino C	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: використано середовище Arduino, оскільки є підтрримка плати ATmega2560 та плати ESP8266, наявні бібліотеки з відкритою ліцензією для периферії.				
2.	Використання операційної системи реального часу (OSPB) для вбудованих систем	FreeRTOS	Наявна	Доступна
		scmRTOS	Наявна	Доступна
		Armbian	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: FreeRTOS, через підтримку архітектури AVR.				
3.	Вимірювання рівня води у баці чистої води.	Датчик поплавкового типу для баку палива катерів або лодок	Наявна	Доступна
		Датчик поплавкового типу для паливного баку автівки.	Наявна	Доступна
		Власний датчик поплавкового типу	Потрібно розробити	Доступна
		Індукційний датчик	Наявна	Доступна

		Сонар як вимірювач відстань від верхньої точки резервуара до площини води.	Наявна	Доступна
		Тензозатчик	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: Датчик поплавкового типу для баку палива катерів або човен як найбільш надійний та з найбільшою серед усіх розглянутих дискретністю.				

5.3 Аналіз ринкових можливостей запуску стартап-проекту

Потенційний ринок не менш важлива складова окрім сильних сторін системи. Аналіз потенційного ринку представлений у таблиці 5.4.

Таблиця 5.4 – Попередня характеристика потенційного ринку стартап-проекту

№ п/п	Показники стану ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	3
2.	Загальна кількість продаж, грн/ум.од.	3 000 000
3.	Динаміка ринку	Позитивна
4.	Наявність обмежень для входу	Існуючі обмеження для входу: 1.Відсутність стандарту безпеки 2.Відсутність стандартизованого керівництва користувача

		3. Велика конкуренція
5.	Специфічні вимоги для стандартизації специфікації	<ul style="list-style-type: none"> - пожежна безпека; - безпека від протікання; - класифікація як побутовий прилад; - безпека від ураження електричним струмом; - класифікація як фермерське знаряддя.
6.	Середня норма рентабельності в галузі, %	70%

Як можна зазначити, базуючись на даних таблиці 5.5, можна зазначити, що якості потенційних клієнтів є задовільними. Проект виявиться для них привабливим, отже потенційно, покупці будуть наявні. Також необхідно фактори загроз та можливостей у таблицях 4.6 та 4.7 відповідно.

Таблиця 5.5 – Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1.	Полив рослини без участі людини	Садівники	Економлення часу з оптимальним капіталовкладенням	Якість виконання завдання. Мінімізація потреби у

				наповнені резервуарів, оптимальний розмір баків.
		Агрокомпанії	Автоматизація процесу поливу зі зменшенням кількості необхідних працівників	Правильні розрахунки при поливі. Можливість групування об'єктів.
2.	Налаштування процесу додавання добрив	Садівники	Необхідність точних розрахунків у визначенні необхідної кількості речовин.	Наявність необхідних добрив у загальному доступі.
		Агрокомпанії	Стандарти використання речовин у якості добрив. Наявні обмеження щодо використання деяких добрив.	Використання оптимальних технічних рішень щодо процесу. Прозорість та можливість коригування вже підрахованих значень.
3.	Налаштування процесу догляду за рослиною у	Агротехніки	Наявність обмеженої кількості визначених сценаріїв.	Відмовостійкість Прозорість та можливість

	відповідності до її особистого життєвого циклу, без прив'язки до календарного часу		Можливість відмови таких систем. Важкість підрахунку таких значень для нових рослин.	коригування значень. Можливість створення нових сценаріїв. Аналіз отриманих результатів.
4.	Зменшення ваги людського фактору у агровиробництві	Агрокомпанії	Наявність необхідності кваліфікованого персоналу. Можливість людської помилки без автоматизації процесів.	Автономна робота системи. Точність підрахунків. Дистанційне управління. Економічні переваги.
5.	Діагностика стану родини у реальному часі	Приватні підприємці	Можливість втраченої інформації при наданні звітів персоналом.	Дистанційне управління. Отримання звітів щодо параметрів у реальному часі.

Таблиця 5.6 – Фактори загроз

№ п/п	Фактор	Зміст загрози	Можлива реакція компанії
1.	Банкрутство	Стартап зазвичай використовує кошти інвесторів для ведення	Спроба розвивати проект за власні кошти.

		діяльності. Венчурні інвестиції мають високі ризики. Інвестор може відмовитися інвестувати проект.	Отримання кредиту.
2.	Відсутність патенту	Для того щоб захистити інтелектуальну власність необхідно отримати патент. Цей процес є довгостроковим, тому він може призупинити вихід на ринок та оприлюднення розробки світу.	Випуск продукції без патенту
3.	Втрачена актуальність	За час розробки проекту він може втратити актуальність за будь-яких причин. Прорахована рентабельність проекту може бути знецінена якщо інша компанія, наприклад, вже почне виробляти такий самий продукт, але нижчої ціни.	Робити маневрування та змінювати сферу діяльності на актуальну з використання розробок з попереднього проекту.
4.	Неспроможність конкуренції	Компанія може вийти на ринок, але, як і у випадку з загрозою втраченої актуальності, може з'явитися компанія, яка робить теж саме за менші гроші не поступаючись якістю.	Вдосконалювати продукт для набуття рентабельності. Рекламувати продукцію.
5.	Крадіжка інтелектуальної	Викрадення конфіденційної інформації на електронних	Звернення до суду для оформлення іску на

	власності	носіях та розголошення комерційних таємниць з метою перешкоджання ведення діяльності компанії або провокування її подальшого банкрутства.	крадіжку інтелектуальної власності. Проведення розслідування.
--	-----------	---	---

Таблиця 5.7 – Фактори можливостей

№ п/п	Фактор	Зміст можливості	Можлива реакція компанії
1.	Зворотній зв'язок з клієнтами	Оптимізація витраченого часу на розробку модулів. Знаходження слабких місць та їх ліквідація.	Комунікація з клієнтами. Розуміння потреб клієнтів.
2.	Додавання нових функціональних модулів.	Розробка, тестування та впровадження нових можливостей у продукт.	Збільшення зручності системи та її клієнтоорієнтованості. Наймання нового персоналу. Розширення виробництва

Таблиця 5.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути
---------------------------	---	--

середовища		конкурентоспроможною)
1. Тип конкуренції – чиста	Ринок пристроїв представляє собою рівних конкурентів, які надають різні можливості та характеристики	Компанія має розвивати відмінні характеристики та дотримуватись практик якісного виробництва.
2. За рівнем конкурентної боротьби – національна	Потенційними клієнтами є приватні особи аграрні підприємства на всій території України.	Збільшення обсягів реклами та згадувань. Аналіз поведінки існуючих користувачів для забезпечення кращого досвіду у клієнтів.
3. За галузевою ознакою – внутрішньогалузева	Усі конкуренти знаходяться в рамках однієї галузі.	Конкуренти зазвичай використовують однакові технології і підходи у розробці.
4. Конкуренція за видами товарів: товарно-видова	Конкуренція відбувається між різними виробниками систем для оптимізації процесу зрощування.	Більш детальний аналіз досвіду інших компаній у цій галузі. Необхідність у розробці унікального функціоналу.
5. За характером конкурентних переваг : нецінова	Значну роль грає додавання нового функціоналу.	Розробка та аналіз можливостей впровадження та тестування нового функціоналу.
6. За інтенсивністю : марочна	Для просування компанії важливим є	Визначення цільової аудиторії та збільшення обсягу реклами для

	впізнаваність бренду	неї.
--	----------------------	------

Таблиця 5.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Клієнти	Товари-замінники
	Grobo	Наразі потенційні конкуренти відсутні	Покупка роздріб, наявність відомого впізнаваного бренду	Розрізнені підсистеми для окремих параметрів з вимогами збірки користувачем вручну
	Leaf			
Висновки:	Наразі є декілька прямих конкурентів, але наявний функціонал з розробленою системою дещо відрізняється	Потенційних конкурентів у даний час немає. Перешкоди входу на ринок відсутні.	Клієнти бажають співпрацювати з вже впізнаними брендами, що спонукає розвиток у цьому напрямку.	Обмеження відсутні

SWOT-аналіз наведено у таблиці 5.12. Отримано наступні сильні сторони виходячи зі SWOT-аналізу: унікальний функціонал, зручний інтерфейс, власний алгоритм, автономність, надійність.

Таблиця 5.10 – Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування
1.	Інноваційність	Система розроблена з використанням кокосового субстрату. Ця ніша майже не зайнята, оскільки більшість систем розрахована на використання гідропоніки.
2.	Унікальність функціоналу	Додавання функціоналу сценаріїв підвищує гнучкість системи.
3.	Захищеність	Зумовлена протоколами, які використовують обрані технології, які є найбільш оптимальними.
4.	Сучасний дизайн	Дизайн та ергономіка диктують сприйняття людиною продукту та виробляють позитивні асоціації .
5.	Відмовостійкість	При збою у зовнішніх факторах або внутрішніх система гарантує збереження інформації та стійкість до таких випадків.

Таблиця 5.11 – Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з системою						
			- 3	-2	-1	0	+1	+2	+3

1.	Інноваційність	15			+				
2.	Унікальність функціоналу	19	+						
3.	Захищеність	17				+			
4.	Сучаснісний дизайн	13					+		
5.	Відмовостійкість	20			+				

У порівняльному аналізі у таблиці 5.11, також зазначені сильні сторони на надано їх оцінку. Виявлено фактори конкурентоспроможності у таблиці 5.10 на основі даних про сильні сторони.

Таблиця 5.12 – SWOT- аналіз стартап-проекту

<p>Сильні сторони:</p> <ul style="list-style-type: none"> - унікальний функціонал; - зручний інтерфейс; - власний алгоритм; - автономність; - надійність; - зручність. 	<p>Слабкі сторони:</p> <ul style="list-style-type: none"> - енергонеефективність; - немає відомого бренду; - може бути поглинута конкурентами.
<p>Можливості:</p> <ul style="list-style-type: none"> - масштабування системи на 1, 2, n рослин та більше; - розширення ринку завойовуванням попиту інших континентів; - удосконалення алгоритму; 	<p>Загрози:</p> <ul style="list-style-type: none"> - дефіцит фінансової підтримки під час виробництва; - поява дешевшого аналога також ж функціоналу; - неправильно порахований попит відділом маркетингу, за рахунок чого дефіцит

- звернути у іншу сферу діяльності .	попиту та банкрутство.
--------------------------------------	------------------------

Таблиця 5.13 – Альтернативи ринкового впровадження стартап-проекту

№ п/п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1.	Вивід продукту на ринок з існуючим дизайном та апаратними можливостями.	Залучення сторонніх спонсорських інвестицій.	2 місяці
2.	Покращення характеристик та вивід на ринок.	Отримання фінансування шляхом процедури попереднього замовлення.	4 місяці
3.	Продаж технології компанії-конкуренту або поглинання.	Отримання коштів за рахунок угоди купівлі продажу.	1 місяць

5.4 Розроблення ринкової стратегії проекту

Таблиця 5.14 – Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
-------	--	---	---	--------------------------------------	--------------------------

1.	Агropідприємства	Середня готовність	Середній попит	Збільшення конкуренції	Середня складність
2.	Приватні особи	Висока готовність	Середній попит	Збільшення конкуренції	Середня та нижче середнього
3.	Агротехніки	Висока готовність	Високий попит	Відсутність конкуренції	Низька складність
4.	Державні підприємства	Низька готовність	Низький попит	Висока конкурентність	Висока складність
Які цільові групи обрано: було обрано цільові групи 2, 3.					

Таблиця 5.15 – Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
1.	Отримання фінансування за рахунок введення передзамовлення, вивід бета версії для тестової групи покупців з метою виявлення помилок та покращення продукту за рахунок	Ексклюзивний розподіл	Розробка системи для можливості предзамовлень. Тестування існуючого функціоналу. Наявність тонкого налаштування системи.	Стратегія диференціації

	зворотнього зв'язку			
--	---------------------	--	--	--

Таблиця 5.16 – Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
1.	Проект частково є першопрохідцем	Спершу пріоритетом буде наращення своєї бази клієнтів, яким буде зручно користуватись першою версією. Наступний крок припускає залучення клієнтів у конкурентів.	Компанія буде вміщувати найважливіші характеристики товару конкурента, додаючи нові можливості і новий підхід до цієї проблеми.	Стратегія диференціації

Таблиця 5.17 – Визначення стратегії позиціонування

№ п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного	Вибір асоціацій, які мають формувати комплексну
----------	---	---------------------------------	---	---

			стартап- проекту	позицію власного проекту (три ключових)
1.	Можливість налаштування дистанційно у реальному часі.	Стратегія спеціалізації	Інтерфейс для взаємодії з клієнтом	Дистанційна робота з приладом з будь-якої точки де є мережа інтернет.
2.	Отримання актуальних даних про стан системи.	Стратегія спеціалізації	Використання різних датчиків для отримання інформації	Актуальні дані про стан системи завжди під рукою.
3.	Тонке налаштування параметрів.	Стратегія спеціалізації	Наявність функціоналу сценаріїв	Наявність сценаріїв для налаштування під будь-яку рослину.

5.5 Розроблення маркетингової програми стартап проекту

У таблиці 5.18 описано маркетингову концепцію, визначено ключові переваги, межі встановлення ціни.

Таблиця 5.18 – Визначення ключових переваг концепції потенційного товару

№ п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1.	Можливість налаштування циклів для рослини, відповідно до її	При підрахунку значень враховуються біологічні періоди	Конкуренти у своїх продуктах зазвичай використовують календарний час, на відміну від розроблюваного продукту.

	біологічного часу	рослини	
2.	Наявність пристрою “у коробці” з мінімальною складністю налаштування перед початком роботи.	Система має API для інтегрування інтерфейсу, усі фізичні модулі зібрані з початку.	Конкурентні продукти або мають інтуїтивно зрозумілий інтерфейс та мінімальний функціонал або навпаки. Розроблювана система уособлює у собі обидва параметра.

Таблиця 5.19 – Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Автономна система для зрошування рослин, яка призначена для збільшення продуктивності росту рослин, інтерактивна, з оповіщенням у теплиці.		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	Система автономна, для збільшення продуктивності росту рослин, інтерактивна, з оповіщенням призначила для гроубоксу для особистого використання	М	Тх/Тл/Е
	Система була протестована, вона задовольняє всім необхідним вимогам для того щоб вийти на ринок.		
	Веб-сайт		
III. Товар із підкріпленням	Система призначена для автоматизації зрошування		

Потенційний товар буде захищено від копіювання за рахунок пережигання фьюз-бітів мікроконтролера, що не дозволяють викачати прошивку з останнього, за рахунок моніторингу підключених пристроїв. Вся система працює через інтернет, кожен має свій унікальний ID у прошивці, якщо ж вдається скопіювати прошивку, буде наявно більше одного не унікального пристрою.

Таблиця 5.20 – Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1.	Нижчий відносно розроблюваного продукту.	Нижча відносно аналогів.	Середній рівень доходів	Нижня: 1500 ум. од. Верхня: 2300 ум. од.

Таблиця 5.21 – Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
1.	Клієнти придбають продукт напряму, зазвичай у одиничній кількості.	Виконання отриманого замовлення. Надання гарантійного талону. Супровід клієнта.	Виробник – споживач	Веб-сторінка

Збут відбуватиметься за схемою, вказаною у таблиці 5.21, та підтримуватися на високому рівні завдяки концепціям маркетингових комунікацій, описаних у таблиці 5.22.

Таблиця 5.22 – Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1.	Цільові клієнти у більшості шукають на ринку рішення, яке задовольняє їх специфічні технічні вимоги. Складність налаштування та збірки продукту відштовхує від потенційного	Спеціалізовані форуми, веб-сайти, соціальні мережі.	Розробка іміджу, таргетингова реклама, реклама в соціальних мережах.	Ознайомлення потенційних клієнтів з можливостями та перевагами продукту.	Представлення можливостей та результатів

	<p>придбання. Розроблена система вирішує це питання та має принести влив на ринок даної галузі.</p>				
--	---	--	--	--	--

ВИСНОВКИ

У даній магістерській дисертації було розглянуто, розроблено та реалізовано систему автоматизації життєвих процесів рослини. Проведено аналіз ринку та конкурентів задля виявлення актуальності даної розробки та вимірювання величини попиту.

Під час створення було розглянуто функціонал конкурентів, та покращено користувацький досвід і підхід до створення систем даного типу для отримання найкращого користувацького досвіду у новому форматі.

Створено алгоритм, за яким система має змінювати параметри суміші. На основі структурної, функціональної схеми, алгоритму та технічного завдання спроектовано програмне забезпечення для МК.

Розроблено механізм зберігання сценаріїв у МФК. Створено «захоплюючий портал» для первинного підключення пристрою до мережі Інтернет. Програмне забезпечення готове до завантаження у МК.

Розглянуто технології за допомогою яких був створений даний продукт, та обґрунтовано вибір кожної з них. Надано публічний інтерфейс додатку, за допомогою якого можна долучитись до створення системи керування приладом, розробленим у рамках магістерської дисертації. Елементи змонтовано та підключено. Програмну частину завантажено до МК.

Завдяки наявному обладнанню та проведеним тестам отримано результати роботи системи, що задовольняють потреби. Система підтримує параметри кислотності у границях «задана – 0.5 од» < задана < «задана + 0.5 од», що є допустимим для проходження росту рослини, подає потрібні кількості добрив та кількість чистої води.

Системою пройдено тести на працездатність. У розділі 4 наявні результати тестування системи.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гидропоника – Что и зачем это применяется [Електронний ресурс] – Режим доступу до ресурсу: <https://www.fertilizerdaily.ru/20200421-gidroponika-cto-eto-i-zachem-primenyaetsya/>
2. Ткачук О.П. СІЛЬСЬКОГОСПОДАРСЬКА ЕКОЛОГІЯ / Ткачук О.П., Шкатула Ю.М., Тітаренко О.М.. – Вінниця, 2020. – 542 с.
3. Grobo – Automated Grow Box – Shop Now [Електронний ресурс] – Режим доступу до ресурсу: <https://www.grobo.io/>
4. LEAF: The Automatic Home Grow System [Електронний ресурс] – Режим доступу до ресурсу: <https://www.getleaf.co/>
5. Перспективы роботизации и автоматизации тепличного овощеводства [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: <https://greentalk.ru/topic/13117/>.
6. BC Northern Lights [Електронний ресурс] – Режим доступу до ресурсу: <https://bcnorthernlights.com/>
7. New Age of Agriculture: State of Digital Farming in 2020 [Електронний ресурс] – Режим доступу до ресурсу: <https://industrywired.com/new-age-of-agriculture-state-of-digital-farming-in-2020/>
8. Prevention of heart attack [Електронний ресурс] – Режим доступу до ресурсу: <https://www.hackster.io/colonel/prevention-of-heart-attack-09f726>
9. Что Такое JSON? [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: <https://www.hostinger.com.ua/rukovodstva/chto-takoe-json/>.
10. Защита систем от кибер-атак [Електронний ресурс] – Режим доступу до ресурсу: <https://drive.google.com/file/d/0B9GbRx6dJ9WCQTFEeHBPWGM0aWc/edit>.
11. <https://qastack.ru/electronics/433/programming-microcontrollers-jtag-spi-usb-oh-my>

12. Андрей Курниц. FreeRTOS — операционная система для микроконтроллеров // Компоненты и Технологии (КиТ). — 2011. — № 2, 3, 4, 5, 6, 7, 8, 9, 10. — С. 96-100, 109-114, 96-102, 97-102, 98-104, 23-32, 132-140, 97-104, 93-100.
13. Shafranovich, Y. Common Format and MIME Type for CSV Files, 2005.
14. How to check the strength of your Wi-Fi signal [Электронный ресурс] – Режим доступа до ресурсу: <https://www.businessinsider.com/how-to-check-wifi-signal-strength..>
15. ESP8266 WiFi Captive Portal [Электронный ресурс]. – 2020. – Режим доступа до ресурсу: https://github.com/125K/ESP8266_WiFi_Captive_Portal.
16. DS18B20 – датчик температуры с интерфейсом 1-Wire. Описание на русском языке. [Электронный ресурс]. – 2016. – Режим доступа до ресурсу: <http://mypractic.ru/ds18b20-datchik-temperatury-s-interfejsom-1-wire-opisanie-na-russkom-yazyke.html>.
17. Датчики кислотності розчину [Электронный ресурс] – Режим доступа до ресурсу: https://aliexpress.ru/item/32797327021.html?&sku_id=63876863376.
18. Atmel Studio 7 [Электронный ресурс] – Режим доступа до ресурсу: <http://atmel-studio-doc.s3-website-us-east-1.amazonaws.com/webhelp/GUID-54E8AE06-C4C4-430C-B316-1C19714D122B-en-US-1/index.html?GUID-8F63ECC8-08B9-4CCD-85EF-88D30AC06499>.
19. ПротоколUsart. Исполнительный механизм. [Электронный ресурс] – Режим доступа до ресурсу: <https://studfile.net/preview/5154993/page:4/>.
20. Richard H. B. Embedded C Programming And the Atmel – AVR E-Book / H. B. Richard, D. O. Larry, A. C. Sarah., 2007. – 532 с. – (Cengage Learning).
21. Таблица оптимального уровня pH [Электронный ресурс] – Режим доступа до ресурсу: https://www.umax.pro/?page_id=1184.

